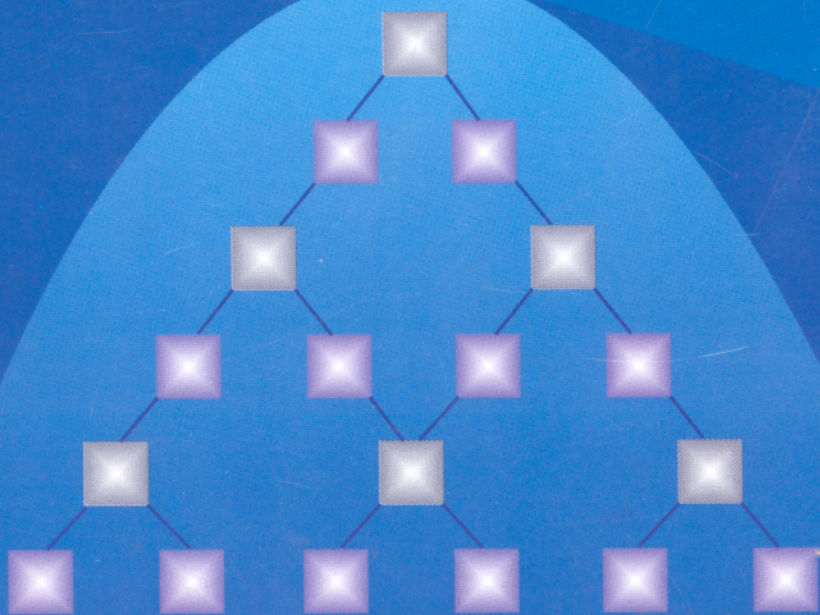


HOÀNG CHÍ THÀNH

ĐỒ THỊ và CÁC THUẬT TOÁN



NHÀ XUẤT BẢN GIÁO DỤC



PGS. TS. HOÀNG CHÍ THÀNH

ĐỒ THỊ và CÁC THUẬT TOÁN

NHÀ XUẤT BẢN GIÁO DỤC

Bản quyền thuộc HEVOBCO – Nhà xuất bản Giáo dục

17 – 2007/CXB/75 – 2217/GD

Mã số: 7B664M7 – DAI

Lời nói đầu

Lý thuyết Đồ thị là một trong những ngành khoa học ra đời khá sớm và có nhiều ứng dụng. Nhờ Lý thuyết Đồ thị mà nhiều bài toán phức tạp, diễn giải dài dòng được mô tả hình học một cách trực quan và cô đọng. Nhiều khái niệm của Lý thuyết Đồ thị được sinh ra từ các vấn đề thực tiễn như: đường đi, chu trình, tập ổn định, nhân, chu số, sắc số, duyệt đồ thị, đường đi ngắn nhất, tâm đồ thị, luồng vận tải, đồ thị phẳng, cây bao trùm, cây biểu thức, cây mã tiền tố tối ưu... Vì vậy, Lý thuyết Đồ thị đã gắn kết nhiều ngành khoa học lại với nhau. Các thuật toán ngắn gọn và lý thú của Lý thuyết Đồ thị đã giúp chúng ta giải quyết rất nhiều bài toán phức tạp trong thực tế.

Cuốn sách này được biên soạn dựa trên các bài giảng mà tác giả đã giảng dạy nhiều năm tại Trường Đại học Khoa học Tự nhiên – Đại học Quốc gia Hà Nội. Cuốn sách cung cấp cho sinh viên, học viên cao học, nghiên cứu sinh các ngành Toán học, Tin học... những kiến thức cơ bản nhất của Lý thuyết Đồ thị cùng các thuật toán tiêu biểu. Các thuật toán được trình bày chi tiết, giúp người học dễ dàng nắm bắt và cài đặt trên máy tính trong các bài toán ứng dụng. Phần bài tập cuối các chương giúp người học tự kiểm tra kiến thức của mình.

Mặc dù đã hết sức cố gắng, song không tránh khỏi những thiếu sót. Tác giả mong nhận được sự góp ý chân tình của bạn đọc gần xa để chất lượng cuốn sách được tốt hơn trong những lần xuất bản sau. Mọi góp ý xin gửi về: Công ty Cổ phần Sách Đại học – Dạy nghề, 25 Hàn Thuyên – Hà Nội.

TÁC GIẢ

Chương 1

ĐẠI CƯƠNG VỀ ĐỒ THỊ

1.1. KHÁI NIỆM ĐỒ THỊ

1.1.1. Định nghĩa đồ thị

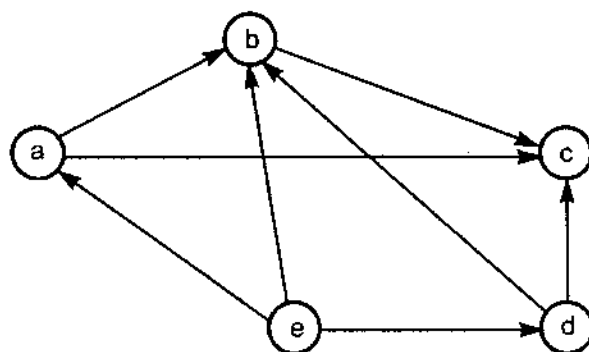
Chúng ta đã nhìn thấy hoặc sử dụng bản đồ các tuyến đường giao thông của một thành phố, sơ đồ tổ chức của một cơ quan, sơ đồ khối tính toán của một thuật toán, sơ đồ một mạng máy tính... Đó là những ví dụ cụ thể về đồ thị.

Đồ thị (*graph*) là một mô hình toán học được ứng dụng trong nhiều lĩnh vực khoa học, kỹ thuật và được định nghĩa như sau:

Định nghĩa 1.1: Đồ thị là một cặp $G = (V, E)$, trong đó:

- 1) V là tập hợp các *đỉnh* (*vertex*);
- 2) $E \subseteq V \times V$ là tập hợp các *cạnh* (*edge*).

Ví dụ 1.1:



Hình 1.1. Đồ thị hữu hạn

Đồ thị G ở Hình 1.1 với tập các đỉnh $V = \{a, b, c, d, e\}$ và tập các cạnh $E = \{(a, b), (a, c), (b, c), (d, b), (d, c), (e, a), (e, b), (e, d)\}$.

Nếu (a, b) là một cạnh của đồ thị thì ta nói rằng đỉnh b kề với đỉnh a ; hai đỉnh a và b kề với cạnh (a, b) .

Trong đồ thị ở Ví dụ 1.1, hai đỉnh b và c kề với đỉnh a ; ba đỉnh a, b và d kề với đỉnh e . Do vậy, có thể định nghĩa đồ thị bằng ánh xạ kề như sau:

Định nghĩa 1.2: Đồ thị là một cặp $G = (V, F)$, trong đó:

- 1) V là tập hợp các đỉnh;
- 2) $F : V \rightarrow 2^V$ được gọi là *ánh xạ kề*.

Ánh xạ kề của đồ thị trong Ví dụ 1.1 được xác định như sau:

$$F(a) = \{b, c\}, F(b) = \{c\}, F(c) = \emptyset, F(d) = \{b, c\} \text{ và } F(e) = \{a, b, d\}.$$

Sự tương đương của hai định nghĩa của đồ thị được thể hiện bằng mệnh đề sau:

$$\forall x, y \in V : (x, y) \in E \Leftrightarrow y \in F(x).$$

Về bản chất, đồ thị là một tập hợp các đối tượng được biểu diễn bằng các đỉnh và giữa các đối tượng này có một quan hệ nhị nguyên biểu diễn bằng các cạnh.

Cặp đỉnh $(x, y) \in E$ không sắp thứ tự được gọi là *cạnh vô hướng*, còn nếu nó có sắp thứ tự thì được gọi là *cạnh có hướng*. Vì thế, đồ thị thường phân thành hai lớp.

Định nghĩa 1.3: Đồ thị chỉ chứa các cạnh vô hướng được gọi là *đồ thị vô hướng*, còn đồ thị chỉ chứa các cạnh có hướng được gọi là *đồ thị có hướng*.

Hiển nhiên, mỗi đồ thị vô hướng có thể biểu diễn bằng một đồ thị có hướng bằng cách thay mỗi cạnh vô hướng bằng hai cạnh có hướng tương ứng.

Định nghĩa 1.4: Đồ thị $G = (V, E)$ được gọi là *đối xứng* nếu:

$$\forall x, y \in V : (x, y) \in E \Leftrightarrow (y, x) \in E.$$

Các đồ thị vô hướng là đối xứng.

Định nghĩa 1.5: Đồ thị $G = (V, E)$ mà mỗi cặp đỉnh được nối với nhau bởi không quá một cạnh được gọi là *đơn đồ thị* (thường gọi tắt là *đồ thị*). Nếu đồ thị có những cặp đỉnh được nối với nhau nhiều hơn một cạnh thì được gọi là *đa đồ thị*.

Có thể biểu diễn hình học cho đồ thị như sau: Trên mặt phẳng, đỉnh được biểu diễn bằng các vòng tròn nhỏ; cạnh vô hướng được biểu diễn bằng đoạn thẳng; cạnh có hướng được biểu diễn bằng đường mũi tên nối hai đỉnh của đồ thị.

Trong giáo trình này, chỉ xét các đồ thị hữu hạn, nghĩa là các đồ thị có tập đỉnh là hữu hạn.

1.1.2. Đường đi và chu trình

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 1.6: Đường đi trong đồ thị là một dãy các đỉnh:

$$\langle x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{k-1}, x_k \rangle$$

sao cho mỗi đỉnh trong dãy (không kể đỉnh đầu tiên) kề với đỉnh trước nó bằng một cạnh nào đó, nghĩa là: $\forall i = 2, 3, \dots, k-1, k : (x_{i-1}, x_i) \in E$.

Ta nói rằng, đường đi này đi từ đỉnh đầu x_1 đến đỉnh cuối x_k . Số cạnh của đường đi được gọi là *độ dài* của đường đi đó.

Đường đi đơn là đường đi mà các đỉnh trên nó khác nhau từng đôi.

Định nghĩa 1.7: Chu trình là một đường đi khép kín (tức là đỉnh cuối của đường trùng với đỉnh đầu của đường). Chu trình thường ký hiệu là:

$$[x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{k-1}, x_k], \text{ trong đó } x_1 = x_k.$$

Để cho gọn, trong ký hiệu của chu trình thường không viết đỉnh cuối:

$$[x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_{k-1}].$$

Khi nói đến một chu trình, cũng không cần xác định đỉnh đầu và đỉnh cuối của chu trình đó.

Chu trình được gọi là *chu trình đơn* nếu các đỉnh trên nó khác nhau từng đôi.

Trong một đồ thị, *đỉnh nút* là đỉnh kề với chính nó. Hai cạnh có ít nhất một đỉnh chung được gọi là *hai cạnh kề nhau*.

Để việc trình bày được ngắn gọn, trong suốt cuốn sách này, ký hiệu n là số đỉnh, m là số cạnh của một đồ thị.

1.1.3. Đồ thị con và đồ thị riêng

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 1.8:

1) Đồ thị $G' = (V', E')$ được gọi là *đồ thị con* của đồ thị G nếu:

$$V' \subseteq V \text{ và } E' = E \cap (V' \times V').$$

Đồ thị $G'' = (V, E'')$ với $E'' \subseteq E$ được gọi là *đồ thị riêng* của đồ thị G .

Mỗi tập con các đỉnh V' của đồ thị tương ứng duy nhất với một đồ thị con, do vậy để xác định một đồ thị con chỉ cần nêu tập đỉnh của nó. Còn đồ thị riêng là đồ thị giữ nguyên tập đỉnh và bỏ bớt một số cạnh.

1.1.4. Sự đẳng hình của các đồ thị

Sự đẳng hình của hai đồ thị dựa trên sự đẳng cấu của hai tập đỉnh sao cho sự đẳng cấu ấy bảo toàn được các cạnh của đồ thị.

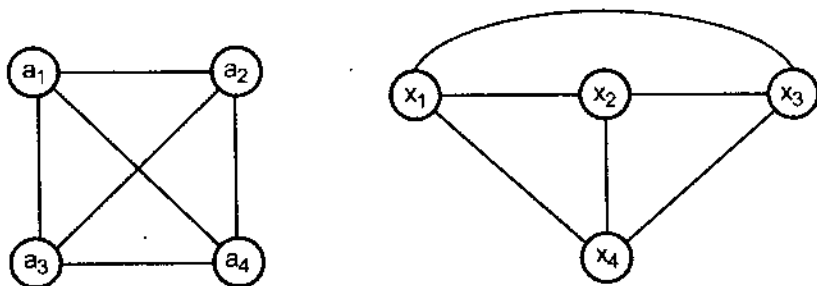
Định nghĩa 1.9: Hai đồ thị $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ được gọi là *đẳng hình* nếu tồn tại một song ánh S trên các tập đỉnh $(S: V_1 \rightarrow V_2)$ bảo toàn các cạnh:

$$\forall x, y \in V_1 : (x, y) \in E_1 \Leftrightarrow (S(x), S(y)) \in E_2.$$

Chúng ta không phân biệt hai đồ thị đẳng hình với nhau, vì về thực chất chúng chỉ khác nhau về tên gọi của các đỉnh và cách biểu diễn bằng hình vẽ.

Ví dụ 1.2: Hai đồ thị dưới đây (Hình 1.2) là đẳng hình với song ánh:

$$S(a_i) = x_i, i = 1, 2, 3, 4.$$



Hình 1.2. Hai đồ thị đẳng hình

1.1.5. Các cách biểu diễn đồ thị trong máy tính

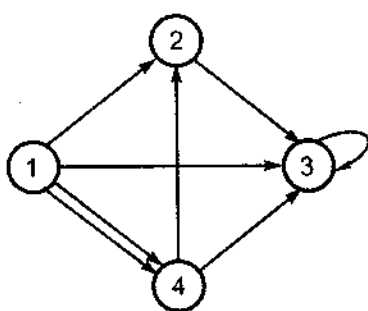
a) Biểu diễn đồ thị bằng ma trận kề

Giả sử $G = (V, E)$ là một đồ thị. Đánh số các đỉnh của đồ thị bằng các số tự nhiên: $1, 2, \dots, n$. Xây dựng ma trận vuông biểu diễn đồ thị như sau:

Ma trận vuông $A_{n \times n}$ được gọi là *ma trận kề* của đồ thị G nếu: $\forall i, j \in V, A[i, j] = d$, trong đó d là số cạnh nối đỉnh i với đỉnh j trong G .

Dễ thấy rằng, đồ thị G là đối xứng khi và chỉ khi ma trận kề A là đối xứng.

Ví dụ 1.3: Ma trận kề của đa đồ thị có hướng.



$$A = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Hình 1.3. Đồ thị có hướng và ma trận kề tương ứng

Cách biểu diễn đồ thị bằng ma trận kề cho kết quả sau:

Định lý 1.1: Phần tử ở hàng i và cột j của ma trận lũy thừa A^k là số các đường đi khác nhau có độ dài k nối đỉnh i với đỉnh j trong đồ thị G .

Chứng minh: Chứng minh bằng quy nạp theo độ dài k của đường đi. Thật vậy:

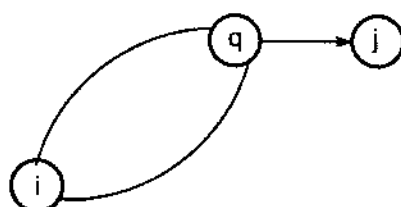
$k = 1$: Suy từ chính định nghĩa của ma trận kề.

$(k) \Rightarrow (k+1)$: Ký hiệu $A = [a_{ij}]$, $A^k = [b_{ij}]$,

$$C = A^k \cdot A = [c_{ij}],$$

khi đó:

$$c_{ij} = \sum_{q=1}^n b_{iq} a_{qj}.$$



Hình 1.4. Các đường đi từ đỉnh i đến đỉnh j qua đỉnh q

Với q bất kỳ ($1 \leq q \leq n$), theo giả thiết quy nạp b_{iq} là số đường đi từ đỉnh i đến đỉnh q có độ dài k . Nếu $a_{qj} = 0$ thì không có cạnh từ q đến j , do đó cũng không có đường đi từ i đến j qua q với độ dài $k + 1$.

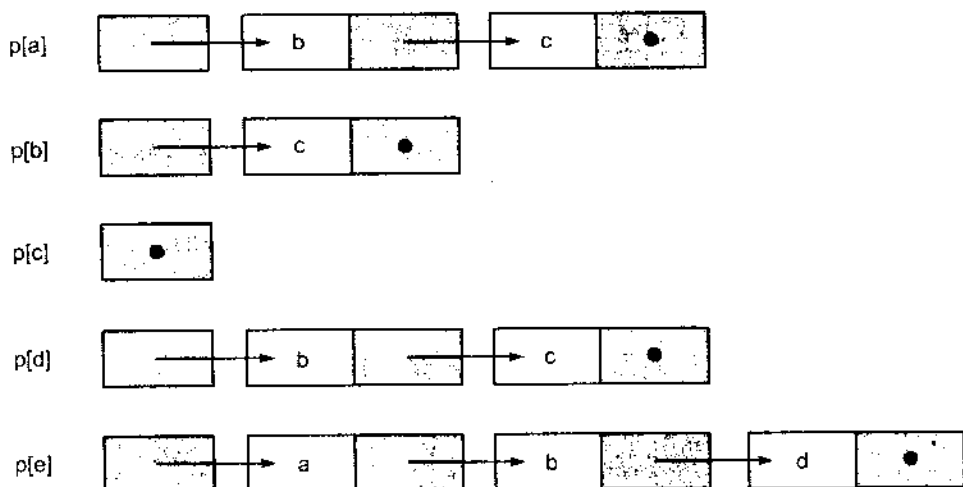
Nếu $a_{qj} = d \geq 1$ thì có cạnh đi từ q đến j . Do đó có các đường đi từ i đến j qua q với độ dài $k + 1$, mà số các đường đi đó chính là $d \cdot b_{iq}$.

Vậy tính c_{ij} theo tổng trên, ta sẽ có tất cả các đường đi từ i đến j với độ dài $k + 1$. Định lý đã được chứng minh. ■

b) Biểu diễn đồ thị bằng danh sách kề

Với mỗi đỉnh của đồ thị, ta xây dựng một danh sách móc nối chứa các đỉnh kề với đỉnh này. Danh sách này được gọi là *danh sách kề*. Một đồ thị được biểu diễn bằng một mảng các danh sách kề.

Ví dụ 1.4: Biểu diễn mảng các danh sách kề của đồ thị G trong Ví dụ 1.1.



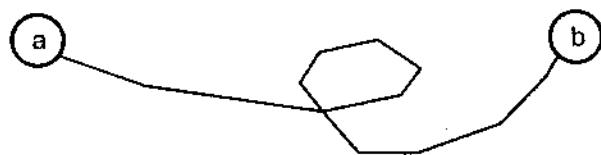
Hình 1.5. Mảng các danh sách kế biểu diễn đồ thị trong Ví dụ 1.1

1.2. MỘT SỐ TÍNH CHẤT VỀ ĐƯỜNG ĐI TRÊN ĐỒ THỊ

Trong phần này, xét một số tính chất của đường đi nối hai đỉnh trong một đồ thị cũng như sự tồn tại của chúng.

Định lý 1.2: Giả sử đồ thị G có n đỉnh. Tồn tại đường đi từ đỉnh a đến đỉnh b trên đồ thị G khi và chỉ khi tồn tại một đường đi từ a đến b trên đồ thị này với độ dài không vượt quá $n - 1$.

Chứng minh: Giả sử có đường đi từ đỉnh a tới đỉnh b , có thể chọn: $\langle a = x_1, x_2, \dots, x_k = b \rangle$ là đường đi có độ dài ngắn nhất.



Hình 1.6. Một đường đi từ đỉnh a đến đỉnh b

Độ dài của đường đi là $k - 1$. Nếu $k - 1 \leq n - 1$ thì định lý được chứng minh.

Nếu ngược lại, $k - 1 > n - 1$, nghĩa là $k > n$, thì trong dãy đỉnh của đường đi sẽ có ít nhất hai đỉnh trùng nhau, chẳng hạn: $x_i = x_j$. Khi đó:

$$\langle a = x_1, x_2, \dots, x_i, x_{j+1}, \dots, x_k = b \rangle$$

cũng là đường đi từ a tới b nhưng với độ dài ngắn hơn. Suy ra mâu thuẫn với giả thiết của đường đi ngắn nhất. Định lý được chứng minh. ■

Xét bài toán đường đi trên đồ thị sau:

Bài toán: Cho đồ thị G và hai đỉnh a, b thuộc G . Có hay không một đường đi từ đỉnh a đến đỉnh b trên đồ thị G ?

Dựa vào Định lý 1.1 và 1.2, xây dựng thuật toán sau đây để trả lời cho bài toán trên.

Thuật toán 1.1:

- 1) Xây dựng ma trận kề A cho đồ thị G .
- 2) Tính ma trận tổng các lũy thừa $T = A^1 + A^2 + \dots + A^{n-1}$.
- 3) Nếu $T[a, b] \geq 1$ thì kết luận là có đường đi từ đỉnh a đến đỉnh b , ngược lại thì kết luận là không có.

Chú ý:

1. Ma trận tổng T còn được gọi là *bao đóng bắc cầu* của ma trận kề A .

2. Các phần tử của ma trận T có thể rất lớn, hơn nữa chúng ta chỉ quan tâm đến tính chất khác 0 của các phần tử, nên có thể xem ma trận kề A như ma trận logic và trong phép nhân ma trận ta thay các phép toán số học $+$, $\times$ bằng các phép toán logic OR và AND. Khi đó, dùng thuật toán Warshall sau đây để tính ma trận bao đóng bắc cầu logic AS. Các phần tử logic của ma trận AS cho biết có hay không đường đi giữa tất cả các cặp đỉnh của đồ thị đã cho.

Thuật toán 1.2 (Warshall):

Dữ liệu: Ma trận kề logic A của đồ thị G .

Kết quả: Ma trận bao đóng bắc cầu logic AS.

```

1  Begin
2  for i := 1 to n do
3    for j := 1 to n do AS[ i, j] := A[ i, j] ;
4  for k := 1 to n - 1 do
5    for i := 1 to n do
6      for j := 1 to n do
7  if !AS[ i, j] then AS[ i, j] :=AS[ i, k] and
AS[ k, j]
8  End.

```

Hiển nhiên, thuật toán có độ phức tạp là $O(n^3)$.

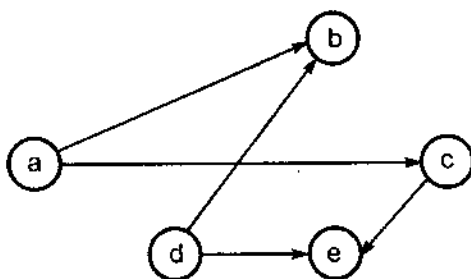
1.3. BẬC CỦA ĐỈNH VÀ TÍNH LIÊN THÔNG CỦA ĐỒ THỊ

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 1.10:

1. Hai đỉnh của đồ thị G được gọi là *liên thông* nếu trên đồ thị có đường đi vô hướng nối chúng với nhau.
2. Đồ thị được gọi là *liên thông* nếu mọi cặp đỉnh của đồ thị đều liên thông với nhau.

Ví dụ 1.5: Đồ thị liên thông.



Hình 1.7. Một đồ thị liên thông

Quan hệ liên thông trên tập đỉnh là một quan hệ tương đương. Nó tạo nên một phân hoạch trên tập các đỉnh. Mỗi lớp tương đương của

quan hệ này được gọi là một *mảng liên thông* (hay *thành phần liên thông*) của đồ thị.

Chú ý rằng:

1. Mỗi mảng liên thông của một đồ thị là một đồ thị con không rỗng liên thông.
2. Hai mảng liên thông khác nhau thì không giao nhau. Do vậy, hai đỉnh ở hai mảng liên thông khác nhau thì không liên thông với nhau.
3. Hợp các mảng liên thông cho ta đồ thị ban đầu.

Ký hiệu p là số mảng liên thông của một đồ thị.

Ta gọi *bậc* của một đỉnh là số cạnh kề với đỉnh đó, và thường ký hiệu là $r(a)$ là bậc của đỉnh a trong đồ thị G .

Định lý 1.3: Tổng tất cả các bậc của các đỉnh trong một đồ thị bằng hai lần số cạnh của đồ thị đó.

Chứng minh: Tính bậc của các đỉnh: Mỗi đỉnh thuộc một cạnh nào đó thì bậc của nó tăng thêm 1. Mà mỗi cạnh thì có hai đỉnh. ■

Hệ quả 1.1: Số đỉnh có bậc lẻ trong một đồ thị phải là một số chẵn.

Hệ quả 1.2: Nếu đồ thị G có đúng hai đỉnh bậc lẻ thì hai đỉnh đó phải liên thông với nhau.

Chứng minh: Giả sử hai đỉnh đó là a và b . Xét mảng liên thông G' chứa a . Bậc của mỗi đỉnh trong G' bằng bậc của đỉnh đó trong G .

Nếu $b \notin G'$ thì trong G' chỉ có một đỉnh bậc lẻ, trái với Hệ quả 1.1. Vậy b phải thuộc mảng liên thông G' chứa a . ■

Định lý 1.4: Đồ thị G có n đỉnh. Nếu bậc của mỗi đỉnh trong G không nhỏ hơn $\frac{n}{2}$ thì đồ thị G liên thông.

Chứng minh: Chứng minh bằng phản chứng.

Giả sử đồ thị G không liên thông. Khi đó, có ít nhất hai đỉnh a và b nằm trong hai mảng liên thông khác nhau. Vậy thì:

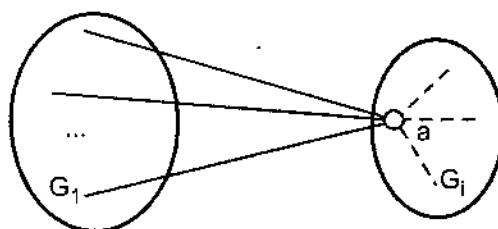
$$n \leq r(a) + r(b) \leq n - 2.$$

Suy ra điều mâu thuẫn. ■

Định lý 1.5: Giả sử đồ thị G có n đỉnh, m cạnh, p mảng liên thông và không có đỉnh nút. Thế thì:

$$m \leq \frac{(n-p)(n-p+1)}{2}.$$

Chứng minh: Giả sử mảng liên thông G_i có n_i đỉnh, $n_i \geq 1$.



Hình 1.8. Cách dồn đỉnh cho mảng G_i

Không mất tính tổng quát có thể xem G_1 là mảng có nhiều đỉnh nhất. Khi đó có thể "dồn" các đỉnh cho mảng G_1 mà không làm thay đổi số đỉnh, số cạnh và số mảng liên thông của đồ thị cho đến khi $n_2 = n_3 = \dots = n_p = 1$.

Thật vậy, giả sử còn mảng G_i mà $n_1 \geq n_i \geq 2$. Chọn a là một đỉnh của G_i sao cho nếu bỏ đỉnh a và các cạnh kề với nó thì phần còn lại vẫn liên thông. Giả sử a được nối với k đỉnh trong G_1 . Hiển nhiên $1 \leq k \leq n_i - 1 < n_1$. Chọn k đỉnh bất kỳ trong mảng G_1 và:

- Thêm k cạnh mới nối a với các đỉnh đã chọn trong G_1 ;
- Xoá bỏ k cạnh nối a với các đỉnh ở trong G_i .

Đỉnh a liên thông với đỉnh trong G_1 nên thuộc vào mảng G_1 . Ta được một đồ thị mới với số đỉnh, số cạnh, số mảng liên thông không thay đổi vì mảng G_i bớt a và k cạnh vẫn còn ít nhất một đỉnh, G_1 thêm đỉnh a và k cạnh mới.

Thực hiện phép "đồn" trên cho đến khi:

$$n_1 = n - p + 1, n_2 = n_3 = \dots = n_p = 1 \text{ và } G_1 \text{ có } m \text{ cạnh.}$$

Vậy $m = \text{số cạnh của } G_1 \leq \binom{n-p+1}{2}$, nghĩa là:

$$m \leq \frac{(n-p)(n-p+1)}{2}.$$

Kết thúc chứng minh định lý. ■

Hệ quả 1.3: Đồ thị G có n đỉnh và có số cạnh $m > \frac{(n-2)(n-1)}{2}$

thì G là liên thông.

Chứng minh: Theo Định lý 1.5 thì:

$$m \leq \frac{(n-p)(n-p+1)}{2},$$

cho nên:
$$\frac{(n-2)(n-1)}{2} < m \leq \frac{(n-p)(n-p+1)}{2}.$$

Bất đẳng thức trên chỉ thỏa mãn khi $p = 1$. Vậy đồ thị G là liên thông. ■

Đồ thị được gọi là *đầy đủ* nếu hai đỉnh bất kỳ đều có cạnh nối. Thường ký hiệu K_n là đồ thị vô hướng đầy đủ n đỉnh.

Trong đồ thị đầy đủ K_n , mỗi đỉnh đều có bậc là $n - 1$ và đồ thị là liên thông. Hai đỉnh bất kỳ được nối với nhau bằng một đường đi ngắn nhất có độ dài bằng 1, đó chính là cạnh nối hai đỉnh ấy.

Một số tính chất:

1) Đồ thị vô hướng với n đỉnh ($n \geq 3$), không có đỉnh nút và bậc của mỗi đỉnh đều không nhỏ hơn 2 luôn có chu trình đơn.

Xét đường đi đơn cực đại $\langle x_1, x_2, \dots, (y), \dots, x_k \rangle$. Khi đó đỉnh $y \neq x_2$ kề với x_1 phải nằm trên đường đi. Từ đó ta có một chu trình $[x_1, \dots, y]$.

2) Đồ thị vô hướng với n đỉnh ($n \geq 4$) và bậc của mỗi đỉnh đều không nhỏ hơn 3 luôn có chu trình đơn độ dài chẵn.

Xét đường đi đơn cực đại $\langle x_1, x_2, \dots (y_1), \dots (y_2), \dots, x_k \rangle$. Khi đó các đỉnh $y_1, y_2 \neq x_2$ kề với x_1 phải nằm trên đường đi. Từ đó ta có ba chu trình: $[x_1, \dots, y_1]$, $[x_1, \dots, y_2]$, $[x_1, y_1, \dots, y_2]$. Nếu hai chu trình đầu có độ dài lẻ thì chu trình thứ ba có độ dài chẵn.

3) Đồ thị với n đỉnh ($n \geq 2$) và không có đỉnh nút có ít nhất hai đỉnh cùng bậc.

– Nếu có đỉnh bậc 0 thì bậc của các đỉnh có thể là: $0, 1, 2, \dots, n-2$.

– Nếu có đỉnh bậc $n-1$ thì bậc của các đỉnh có thể là: $1, 1, 2, \dots, n-1$.

Suy ra số các bậc khác nhau $\leq n-1$. Vậy có ít nhất hai bậc trùng nhau.

4) Nếu đồ thị với n đỉnh ($n \geq 4$) có đúng hai đỉnh cùng bậc thì hai đỉnh này không thể đồng thời có bậc 0 hoặc bậc $n-1$.

Phản chứng: Loại hai đỉnh cùng bậc 0 hay $n-1$ này, ta được đồ thị G' có hai đỉnh cùng bậc. Hai đỉnh này cũng cùng bậc trong G . Mâu thuẫn với giả thiết.

5) Một đồ thị n đỉnh ($n \geq 4$) mà cứ bốn đỉnh tùy ý thì có ít nhất một đỉnh kề với ba đỉnh còn lại, thì có số đỉnh bậc $n-1$ không ít hơn $n-3$.

– Nếu đồ thị G đầy đủ hoặc chỉ có duy nhất hai đỉnh không kề nhau, thì trong G có ít nhất $n-2$ đỉnh bậc $n-1$.

– Nếu đồ thị G có ba đỉnh không kề nhau là a, b, c . Các đỉnh khác phải kề nhau và kề với 3 đỉnh trên. Do vậy, số đỉnh có bậc $n-1$ là $n-3$.

BÀI TẬP

- 1.1. Có bao nhiêu cạnh trong một đồ thị có 10 đỉnh, mỗi đỉnh có bậc là 5?
- 1.2. Liệu có tồn tại hay không một đơn đồ thị có 6 đỉnh với bậc của các đỉnh là: 1, 2, 3, 4, 5, 6?
- 1.3. Giả sử tập $X = \{1, 2, 3, \dots, n\}$. Hỏi:
 - a) Có bao nhiêu đơn đồ thị không có đỉnh nút nhận X là tập các đỉnh?
 - b) Có bao nhiêu đơn đồ thị có đỉnh nút nhận X là tập các đỉnh?
 - c) Có bao nhiêu đa đồ thị mà mỗi cặp đỉnh được nối không quá 2 cạnh nhận X là tập các đỉnh?
- 1.4. Cho một tập hợp X có n phần tử. Xây dựng đồ thị vô hướng mà mỗi đỉnh của nó là một tập con của tập X . Hai đỉnh được nối với nhau bằng một cạnh khi và chỉ khi các tập con tương ứng với chúng không giao nhau.
 - a) Đồ thị trên có bao nhiêu đỉnh, bao nhiêu cạnh? Xét tính liên thông của đồ thị trên.
 - b) Nếu bỏ tập con rỗng của X đi thì kết quả sẽ như thế nào?
- 1.5. Cho một tập hợp X có n phần tử. Xây dựng đồ thị vô hướng mà mỗi đỉnh của nó là một tập con thực sự của tập X . Hai đỉnh được nối với nhau bằng một cạnh khi và chỉ khi các tập con tương ứng với chúng giao nhau. Hỏi đồ thị trên có bao nhiêu đỉnh, bao nhiêu cạnh? Xét tính liên thông của đồ thị trên.
- 1.6. Tìm số đỉnh của một đồ thị có 15 cạnh, ba đỉnh có bậc 4 và các đỉnh còn lại có bậc 3.
- 1.7. Xây dựng thuật toán tính bậc các đỉnh của một đồ thị khi biết biểu diễn ma trận kề của nó.
- 1.8. Xây dựng thuật toán tìm biểu diễn danh sách kề của một đồ thị khi biết biểu diễn ma trận kề của đồ thị này.

1.9. Cho đồ thị $G = (V, E)$ với :

$$V = \{1, 2, \dots, n\}, n \geq 2$$

và $E = \{(i, j) \mid 1 \leq i, j \leq n, i \neq j, i + j \text{ là số chẵn}\}.$

a) Chứng minh rằng, đồ thị G không liên thông và hãy xác định số mảng liên thông của đồ thị này.

b) Nếu thay $i + j$ là số lẻ thì kết quả sẽ như thế nào?

1.10. Trong dịp Tết Nguyên đán, 9 sinh viên cùng lớp về quê ăn Tết. Họ hứa với nhau là mỗi người phải gửi thiệp chúc Tết tới 5 người trong số họ. Chứng minh rằng, có một người không gửi thiệp chúc Tết cho người đã gửi thiệp chúc Tết mình.

1.11. Đồ thị G có n đỉnh và bậc của mỗi đỉnh không nhỏ hơn $\frac{n}{2}$.

Chứng minh rằng, nếu bỏ đi một đỉnh nào đó cùng các cạnh kề với nó thì đồ thị còn nhận được vẫn liên thông.

1.12. Trong một lớp học có 41 sinh viên, người ta nhận thấy cứ trong 6 em bất kỳ luôn có hai em sinh cùng một tháng. Hỏi rằng, có hay không 9 em trong lớp sinh cùng một tháng.

1.13. Trong một lớp học do UNESCO tổ chức tại Tp. Hồ Chí Minh, người ta nhận thấy cứ 8 học viên thì luôn có ít nhất 3 em có cùng một quốc tịch. Hỏi có nhiều nhất bao nhiêu quốc gia tham gia lớp học này?

1.14. Bản đồ giao thông xe buýt của Hà Nội được biểu diễn bằng một đồ thị với tập đỉnh là các điểm đỗ, hai đỉnh có cạnh nối với nhau nếu có xe buýt đi từ đỉnh này tới đỉnh kia. Một khách du lịch đến thăm Thủ đô muốn đi xe buýt từ điểm đỗ i tới điểm đỗ j . Em có thể trả lời cho vị khách đó là đi được hay không (có thể đi liên tuyến)?

1.15. Tìm ba cặp đồ thị, trong từng cặp hai đồ thị đẳng hình với nhau.

1.16. Bình phương của đồ thị định hướng $G = (V, E)$ là đồ thị $G^2 = (V, E^2)$ mà $(u, w) \in E^2$ khi và chỉ khi có đỉnh nào đó

$v \in V$ để $(u, v) \in E$ và $(v, w) \in E$. Lập thuật toán tìm đồ thị G^2 nếu biết ma trận kề A của đồ thị G .

- 1.17.** Ma trận liên thuộc của đồ thị định hướng $G = (V, E)$ là ma trận B cấp $|V| \times |E|$ được định nghĩa như sau:

$$b_{ij} = \begin{cases} -1 & , \text{nếu đỉnh } i \text{ là đỉnh đầu của cạnh } j; \\ 1 & , \text{nếu đỉnh } i \text{ là đỉnh cuối của cạnh } j; \\ 0 & , \text{ngược lại.} \end{cases}$$

Hỏi các thành phần của ma trận tích $B.B^T$ biểu diễn cái gì? (B^T là ma trận chuyển vị của ma trận B).

- 1.18.** Chuyển vị của đồ thị định hướng $G = (V, E)$ là đồ thị định hướng $G^T = (V, E^T)$, mà $E^T = \{(v, u) \in V \times V \mid (u, v) \in E\}$. Thế thì G^T chính là đồ thị G với tất cả các cạnh quay ngược lại. Hãy xây dựng thuật toán tìm đồ thị G^T khi biết biểu diễn danh sách kề của G .

Chương 2

HÀM GRUNDY TRÊN ĐỒ THỊ

Hàm Grundy là một hàm toán học xây dựng trên đồ thị, do P.M. Grundy đề xuất để nghiên cứu một số tính chất lý thú của đồ thị.

Trước tiên, ta ký hiệu tập các số nguyên không âm là $\mathbb{N} = \{0, 1, 2, \dots\}$.

2.1. HÀM GRUNDY

Định nghĩa 2.1: Giả sử $G = (V, F)$ là một đồ thị. Hàm $g : V \rightarrow \mathbb{N}$ được gọi là *hàm Grundy* của đồ thị G nếu:

$$\forall x \in V : g(x) = \min \{ \mathbb{N} \setminus g(F(x)) \}.$$

Từ định nghĩa trên, suy ra hai tính chất đặc trưng của hàm Grundy như sau:

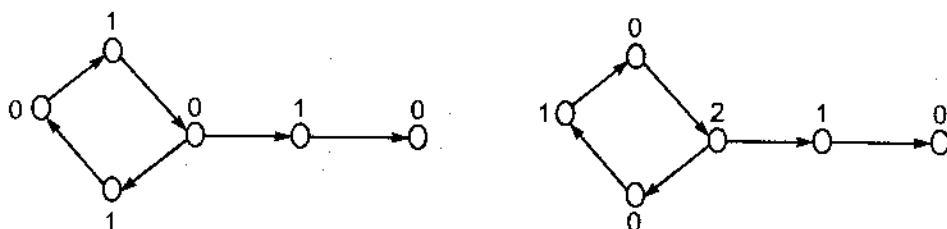
- 1) $\forall x, y \in V$, nếu $y \in F(x)$ thì $g(x) \neq g(y)$.
- 2) $\forall u \in \mathbb{N}, u < g(x) : u \in g(F(x))$; nghĩa là: $\exists y \in F(x)$ để $g(y) = u$.

Tính chất 1) chỉ ra $g(x)$ không nằm trong tập giá trị của g trên $F(x)$, và tính chất 2) khẳng định $g(x)$ là số nguyên không âm bé nhất trong số các số không thuộc tập giá trị của hàm g trên $F(x)$.

Từ định nghĩa của hàm Grundy, ta có một số nhận xét sau:

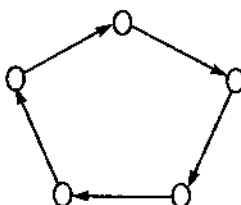
1. Đồ thị có đỉnh nút thì không thể có hàm Grundy.
2. Nếu $F(x) = \emptyset$ thì $g(x) = 0$.
3. Tập hợp $\{x \mid x \in V, g(x) = 0\}$ luôn luôn khác rỗng.
4. $\forall x \in V : g(x) \leq |F(x)|$, nghĩa là hàm Grundy nhận giá trị không lớn.

Ví dụ 2.1: Hàm Grundy có thể không duy nhất.



Hình 2.1. Đồ thị có hai hàm Grundy

Ví dụ 2.2: Hàm Grundy có thể không tồn tại.



Hình 2.2. Đồ thị không có hàm Grundy

Vậy, với điều kiện nào thì một đồ thị có hàm Grundy?

Định lý 2.1: Đồ thị G không có chu trình thì có duy nhất một hàm Grundy.

Chứng minh: Không mất tính tổng quát, có thể giả thiết đồ thị G liên thông. Xây dựng hai dãy tập con các đỉnh: $V_0, V_1, \dots$ và $P_0, P_1, \dots$ lần lượt như sau:

$$V_0 = V,$$

$$P_0 = \{x \mid F(x) = \emptyset\}.$$

Tập P_0 không rỗng. Vì nếu P_0 rỗng, có nghĩa là mọi đỉnh trong V luôn có đỉnh kế. Khi đó xuất phát từ một đỉnh có thể tạo một đường đi dài tùy ý. Điều này là vô lý vì trong G không có chu trình.

$$V_1 = V_0 \setminus P_0,$$

$$P_1 = \{x \mid x \in V_1 \wedge F(x) \subseteq V \setminus V_1\};$$

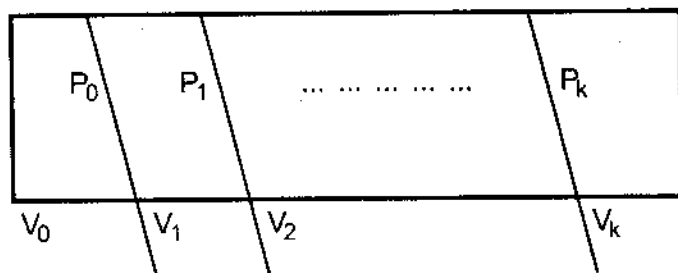
$$V_2 = V_1 \setminus P_1, \dots$$

.....

$$P_i = \{x \mid x \in V_i \wedge F(x) \subseteq V \setminus V_i\}, \text{ với } i \geq 2,$$

$$V_{i+1} = V_i \setminus P_i.$$

Chú ý: Nếu P_k rỗng thì V_k cũng rỗng, nghĩa là đã vét hết các đỉnh của đồ thị. Thật vậy, giả sử ngược lại là P_k rỗng, nhưng V_k không rỗng, khi đó với mỗi $x \in V_k$ sẽ có $y \in F(x)$ để $y \notin V \setminus V_k$, nghĩa là $y \in V_k$. Vậy với mọi đỉnh trong V_k luôn có đỉnh kề cũng thuộc V_k . Khi đó xuất phát từ một đỉnh trong V_k có thể tạo ra một đường đi dài tùy ý. Điều này là vô lý vì đồ thị G không có chu trình.



Hình 2.3. Cách xây dựng dãy tập con $P_0, P_1, \dots, P_k$

Bây giờ xây dựng hàm $g : V \rightarrow \mathbb{N}$ như sau:

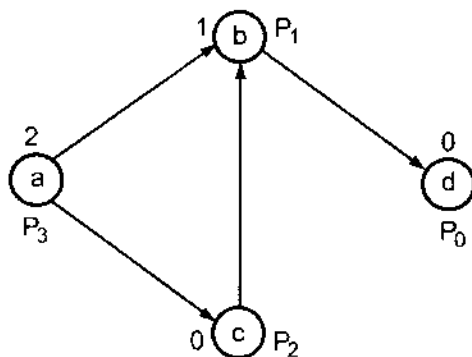
- Với mọi $x \in P_0$ ta đặt $g(x) = 0$.
- Với mỗi $x \in P_1$, nghĩa là $x \notin P_0$ và $F(x) \subseteq P_0$, do hàm g đã được xác định trên P_0 , nên hàm g tại x được xác định một cách duy nhất:

$$g(x) = \min\{N \setminus g(F(x))\}.$$

Tiếp tục cách làm trên sẽ lần lượt xác định được giá trị hàm g tại mỗi đỉnh của đồ thị một cách duy nhất.

Định lý được chứng minh và cách chứng minh đã cho ta thuật toán tìm hàm Grundy cho đồ thị phi chu trình. ■

Ví dụ 2.3: Xét đồ thị có hướng trong Hình 2.4 và cách xây dựng hàm Grundy trên nó.



Hình 2.4. Đồ thị và các tập con P_i

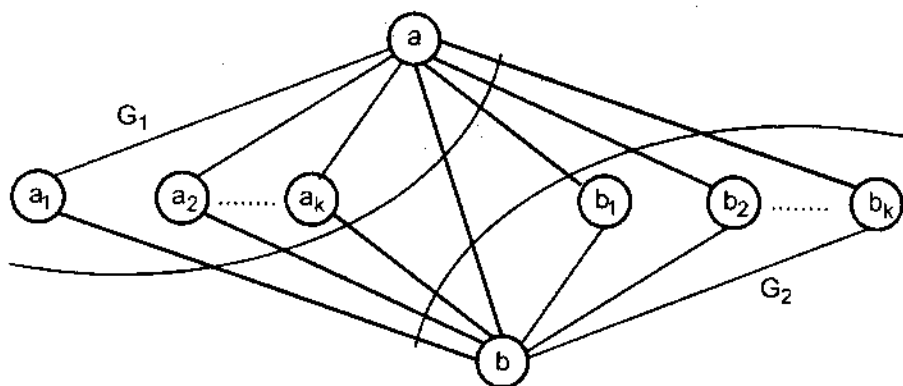
2.2. TỔNG CỦA CÁC ĐỒ THỊ

Cho hai đồ thị dưới dạng ánh xạ kề: $G_1 = (V_1, F_1)$ và $G_2 = (V_2, F_2)$.

Định nghĩa 2.2: Đồ thị $G = (V, F)$ được gọi là *tổng* của G_1 và G_2 , ký hiệu là $G_1 + G_2$ với:

$$1) V = V_1 \times V_2;$$

$$2) (x, y) \in F((a, b)) \Leftrightarrow x = a \wedge y \in F_2(b) \text{ hoặc } x \in F_1(a) \wedge y = b.$$



Hình 2.5. Cách xây dựng đồ thị tổng

Giả sử đồ thị G_1 có hàm Grundy g_1 , đồ thị G_2 có hàm Grundy g_2 . Liệu đồ thị tổng $G_1 + G_2$ có hàm Grundy hay không và mối quan hệ

của nó với các hàm g_1, g_2 thế nào? Để trả lời câu hỏi này, ta đưa ra phép toán d -tổng trên các số nguyên như sau:

Với các số nguyên không âm $u, v \in \mathbb{N}$, biểu diễn chúng dưới dạng nhị phân như sau:

$$u = u_k u_{k-1} \dots u_1 u_0$$

$$v = v_k v_{k-1} \dots v_1 v_0, \text{ với } u_i, v_i \text{ là các chữ số } 0 \text{ hoặc } 1.$$

Có thể xem độ dài biểu diễn của hai số là bằng nhau, nếu không thì thêm các số 0 vô nghĩa vào bên trái số ngắn hơn.

$$\text{Đặt: } w_i = (u_i + v_i) \bmod 2.$$

Định nghĩa 2.3: Số nguyên w có biểu diễn nhị phân là:

$$w_k w_{k-1} \dots w_1 w_0$$

được gọi là d -tổng của u và v và ký hiệu là:

$$w = u \oplus v$$

Chú ý rằng, phép toán này thực hiện giống như câu lệnh gán $w := u \text{ XOR } v$ trong ngôn ngữ lập trình Pascal.

$$\text{Ví dụ 2.4: } 7 \oplus 5 = 2$$

$$12 \oplus 15 = 3$$

Một số tính chất của phép toán d -tổng:

1. Phép toán d -tổng có các tính chất giao hoán và kết hợp:

$$u \oplus v = v \oplus u,$$

$$(u \oplus v) \oplus w = u \oplus (v \oplus w).$$

2. Phép toán d -tổng có đơn vị:

$$u \oplus 0 = 0 \oplus u = u.$$

3. d -tổng của hai số bằng 0 khi và chỉ khi chúng giống nhau:

$$u \oplus v = 0 \Leftrightarrow u = v.$$

Vậy cặp $(\mathbb{N}, \oplus)$ trở thành một nhóm giao hoán.

Định lý 2.2: Nếu g_1 là hàm Grundy của đồ thị G_1 , g_2 là hàm Grundy của đồ thị G_2 thì $g((x, y)) = g_1(x) \oplus g_2(y)$ là hàm Grundy của đồ thị tổng $G = G_1 + G_2$.

Chứng minh: Theo định nghĩa của hàm Grundy, ta cần phải chứng minh:

1. Nếu $(x, y) \in F((a, b))$ thì $g((a, b)) \neq g((x, y))$.
2. Nếu $u \in \mathbb{N}$, $u < g((a, b))$ thì tồn tại $(x, y) \in F((a, b))$ sao cho $g((x, y)) = u$.

Thật vậy, giả sử $(x, y) \in F((a, b))$. Theo định nghĩa của ánh xạ kê F , cần phải xét hai trường hợp sau:

$$1) x = a, y \in F_2(b)$$

Khi đó: $g_2(y) \neq g_2(b)$.

$$g((a, b)) = g_1(a) \oplus g_2(b) = g_1(x) \oplus g_2(b) \neq g_1(x) \oplus g_2(y) = g((x, y)).$$

2) $x \in F_1(a)$, $y = b$: Chứng minh hoàn toàn tương tự. Tính chất 1 được chứng minh.

Bây giờ ta chứng minh tính chất 2.

Giả sử $u \in \mathbb{N}$ và $u < g((a, b))$.

Ký hiệu: $v = g_1(a)$ và $w = g_2(b)$. Ta có:

$$u < v \oplus w.$$

Đặt

$$t = u \oplus v \oplus w.$$

Hiển nhiên

$$t \neq 0 \text{ vì } u \neq v \oplus w.$$

Hơn nữa

$$t \oplus u = u \oplus v \oplus w \oplus u = v \oplus w > u. \quad (*)$$

Xét biểu diễn nhị phân của các số trên:

$$u = \dots u_k = 0 \dots$$

$$v = \dots v_k = 1 \dots$$

$$w = \dots w_k \dots$$

$$t = 0 \dots 01_k \dots$$

Giả sử k là chỉ số của bit đầu tiên bằng 1 trong biểu diễn nhị phân của số t .

Nếu $u_k = 1$ thì $(u_k + t_k) \bmod 2 = 0$. Suy ra: $t \oplus u < u$, trái với mệnh đề (*). Vậy thì $u_k = 0$.

Do bit thứ k của t bằng 1, nên hoặc v_k , hoặc w_k phải bằng 1. Giả sử $v_k = 1$.

Đặt $s = t \oplus v$. Ta có $s < v = g_1(a)$.

Vì g_1 là hàm Grundy của đồ thị G_1 , nên tồn tại $x \in F_1(a)$ sao cho $s = g_1(x)$. Theo định nghĩa của đồ thị tổng thì đỉnh $(x, b) \in F((a, b))$ và $g((x, b)) = g_1(x) \oplus g_2(b) = s \oplus w = t \oplus v \oplus w = u$.

Khi $w_k = 1$ chứng minh hoàn toàn tương tự.

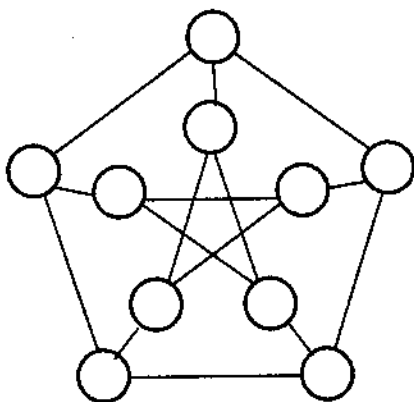
Phần 2) đã chứng minh xong và cũng là kết thúc chứng minh của định lý. ■

Kết quả trên có thể mở rộng cho tổng của nhiều đồ thị.

BÀI TẬP

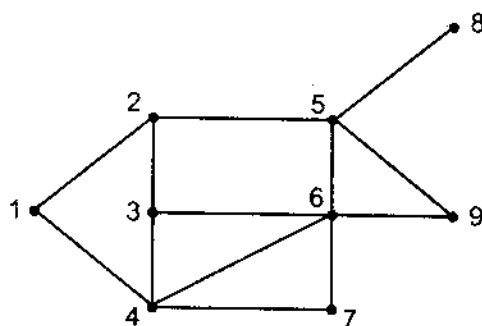
2.1. Xây dựng hàm Grundy cho đồ thị trong Bài tập 1.9.

2.2. Xây dựng tất cả các hàm Grundy cho đồ thị Peterson sau:



Hình 2.6. Đồ thị Peterson 10 đỉnh bậc 3

2.3. Tìm hàm Grundy cho đồ thị được biểu diễn bởi Hình 2.7 sau:



Hình 2.7. Đồ thị vô hướng

- 2.4. Đồ thị đầy đủ n đỉnh K_n có bao nhiêu hàm Grundy? Hãy liệt kê các hàm đó.
- 2.5. Tìm hai đồ thị G_1 và G_2 có hàm Grundy. Xây dựng đồ thị tổng $G_1 + G_2$ và tính hàm Grundy cho đồ thị tổng này.

Chương 3

CÁC TẬP HỢP ĐẶC BIỆT TRÊN ĐỒ THỊ

Trong chương này, chúng ta sẽ nghiên cứu một số tập hợp đặc biệt các đỉnh trên đồ thị. Đó là các tập ổn định trong, tập ổn định ngoài và nhân của một đồ thị.

3.1. TẬP ỔN ĐỊNH TRONG

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 3.1: Tập $B \subseteq V$ được gọi là *tập ổn định trong* của đồ thị G nếu:

$$\forall x \in B : B \cap F(x) = \emptyset.$$

Từ định nghĩa trên thấy rằng, trong một tập ổn định trong không có hai đỉnh nào kề nhau. Hơn nữa, nếu tập B ổn định trong thì tập con $B' \subseteq B$ cũng là tập ổn định trong. Khái niệm ổn định trong không phụ thuộc vào hướng các cạnh của đồ thị.

Ví dụ 3.1 (Gauss): Bài toán tám quân hậu.

Hãy đặt 8 quân hậu vào các ô của một bàn cờ vua sao cho chúng không ăn được lẫn nhau.

Để giải quyết bài toán này, xây dựng đồ thị vô hướng biểu diễn bàn cờ vua như sau: 64 ô của bàn cờ là 64 đỉnh của đồ thị, hai đỉnh x và y có cạnh nối với nhau nếu đó là hai ô mà khi đặt hai quân hậu vào chúng có thể ăn lẫn nhau.

Các ô cần tìm để đặt các quân hậu chính là một tập ổn định trong gồm 8 đỉnh. Bài toán trên có 92 nghiệm, được suy ra từ 12 tập ổn định trong khác nhau là:

{A7, B2, C6, D3, E1, F4, G8, H5}	{A6, B1, C5, D2, E8, F3, G7, H4}	{A5, B8, C4, D1, E7, F2, G6, H3}
{A3, B5, C8, D4, E1, F7, G2, H6}	{A4, B6, C1, D5, E2, F8, G3, H7}	{A5, B7, C2, D6, E3, F1, G4, H8}
{A1, B6, C8, D3, E7, F4, G2, H5}	{A5, B7, C2, D6, E3, F1, G8, H4}	{A4, B8, C1, D5, E7, F2, G6, H3}
{A5, B1, C4, D6, E8, F2, G7, H3}	{A4, B2, C7, D5, E1, F8, G6, H3}	{A3, B5, C2, D8, E1, F7, G4, H6}

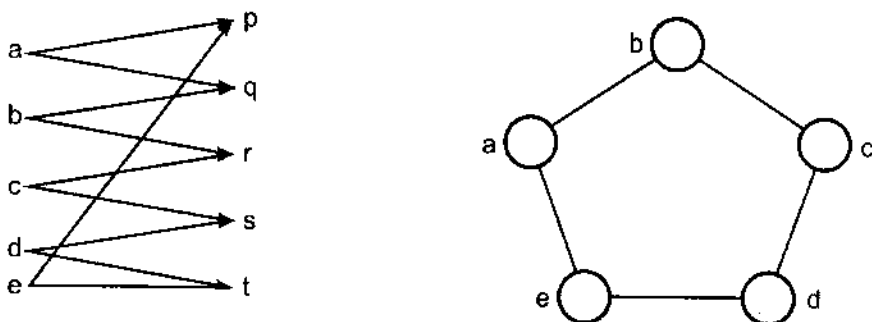
Ví dụ 3.2 (C.E. Shannon): *Bài toán về dung lượng thông tin.*

Giả sử một máy phát có thể truyền đi 5 tín hiệu: a, b, c, d, e . Ở máy thu mỗi tín hiệu có thể cho hai cách hiểu khác nhau như sau:

$$a \rightarrow p, q; b \rightarrow q, r; c \rightarrow r, s; d \rightarrow s, t; e \rightarrow t, p.$$

Hỏi số các tín hiệu nhiều nhất có thể sử dụng để máy thu không bị nhầm lẫn là bao nhiêu?

Xây dựng một đồ thị vô hướng gồm 5 đỉnh a, b, c, d, e . Hai đỉnh là kề nhau nếu chúng biểu thị hai tín hiệu có thể bị nhầm lẫn nhau ở máy thu.



Hình 3.1. Sự nhầm lẫn của các tín hiệu và đồ thị biểu diễn

Khi đó, tập các tín hiệu cần chọn là một trong các tập ổn định trong sau:

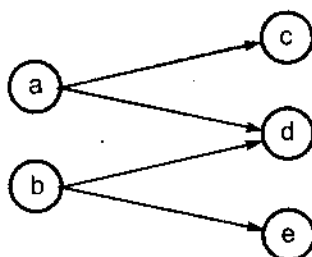
$$\{a, c\}, \{a, d\}, \{b, d\}, \{b, e\}, \{c, e\}.$$

Tập con các đỉnh B được gọi là *tập ổn định trong cực đại* nếu thêm vào bất kỳ đỉnh nào cũng làm mất tính ổn định trong của nó.

Tập B được gọi là *tập ổn định trong lớn nhất* nếu B là tập ổn định trong có nhiều phần tử nhất.

Chú ý: Tập ổn định trong lớn nhất là tập ổn định trong cực đại, nhưng ngược lại thì không đúng.

Ví dụ 3.3:



Hình 3.2. Đồ thị có tập ổn định trong cực đại nhưng không lớn nhất

Các tập $\{a, b\}$ và $\{c, d, e\}$ đều là ổn định trong cực đại.

Lực lượng của tập ổn định trong lớn nhất được gọi là *số ổn định trong* của đồ thị đó, thường ký hiệu số ổn định trong của một đồ thị là u .

Định lý 3.1: Đồ thị G có n đỉnh, bậc lớn nhất của các đỉnh là r .

Khi đó, số ổn định trong của đồ thị $u \geq \frac{n}{r+1}$.

Chứng minh: Giả sử B là tập ổn định trong lớn nhất với u phần tử. Với mỗi đỉnh $y \notin B$ có ít nhất một đỉnh của B kề với y . Vì nếu ngược lại, thì có thể bổ sung y vào B , điều này mâu thuẫn với tính ổn định trong cực đại của B . Từ đó suy ra số cạnh đi ra khỏi B (không kể hướng) lớn hơn hoặc bằng $|V \setminus B| = n - u$.

Mặt khác, số cạnh đó $\leq r.u$. Vậy $r.u \geq n - u$.

Từ đó suy ra: $u \geq \frac{n}{r+1}$. ■

Thuật toán 3.1 (Tìm tập ổn định trong lớn nhất):

- 1) Chọn một đỉnh nào đó của đồ thị.
- 2) Bổ sung dần các đỉnh để được một tập ổn định trong cực đại.
- 3) Nếu ta tìm được một tập ổn định trong có u đỉnh mà mọi tập con $u + 1$ đỉnh đều không là tập ổn định trong, thì kết luận tập

tìm được là tập ổn định trong lớn nhất và u chính là số ổn định trong của đồ thị này.

Trong một đơn vị nào đó, giả sử có quan hệ "*xích mích*" giữa người với người. Thế thì, tập ổn định trong cực đại ở đây được hiểu theo đúng nghĩa xã hội của nó. Đó là một nhóm nhiều người nhất không xích mích với nhau. Để giữ được đoàn kết trong đơn vị thì cần phải xây dựng nhóm này càng lớn càng tốt.

3.2. TẬP ỔN ĐỊNH NGOÀI

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 3.2: Tập $C \subseteq V$ được gọi là *tập ổn định ngoài* của đồ thị G nếu:

$$\forall x \notin C : C \cap F(x) \neq \emptyset.$$

Hay nói một cách khác:

$$\forall x \notin C, \exists y \in C : y \in F(x).$$

Điều này có nghĩa là, từ mỗi đỉnh ở ngoài C luôn có cạnh đi vào C .

Hiển nhiên, nếu C là tập ổn định ngoài thì $C' \supseteq C$ cũng là một tập ổn định ngoài.

Ví dụ 3.4: Hãy đặt 5 quân hậu lên các ô của một bàn cờ vua sao cho chúng kiểm soát được toàn bộ bàn cờ.

Biểu diễn đồ thị cho bàn cờ vua như ở Ví dụ 3.1. Khi đó, các ô cần tìm chính là một tập ổn định ngoài gồm 5 đỉnh.

Một tập nghiệm của bài toán là $\{C6, D3, E5, F7, G4\}$.

Tập C được gọi là *tập ổn định ngoài cực tiểu* nếu bớt đi bất kỳ đỉnh nào của nó cũng làm mất tính ổn định ngoài.

Tập C được gọi là *tập ổn định ngoài bé nhất* nếu C là tập ổn định ngoài có ít phần tử nhất. Lực lượng của tập ổn định ngoài bé nhất được gọi là *số ổn định ngoài* của đồ thị.

Thuật toán 3.2 (Tìm tập ổn định ngoài bé nhất):

Giả sử đồ thị $G = (V, E)$ với tập đỉnh $V = \{a_1, a_2, \dots, a_n\}$.

Thuật toán:

1) Xây dựng ánh xạ $T : V \rightarrow 2^V$ như sau:

$$\forall a \in V, T(a) = \{a\} \cup F^{-1}(a).$$

2) Tìm tập con $C \subseteq V$ có số phần tử ít nhất sao cho $T(C) = V$.

Khi đó, C là một tập ổn định ngoài bé nhất của đồ thị G .

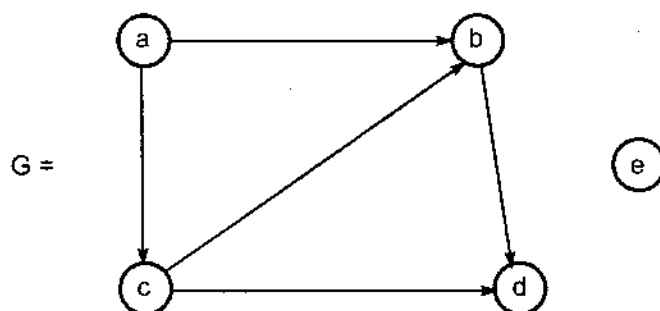
Chú ý: Bước 2) có thể thực hiện nhanh nhờ các nhận xét sau đây:

– Đỉnh cô lập luôn luôn thuộc tập ổn định ngoài bé nhất của đồ thị G , nghĩa là đỉnh cô lập phải được giữ lại.

– Nếu tập con D các đỉnh có tập con C mà: $T(D) \subseteq T(C)$ thì bỏ không xét tập D này.

– Thực hiện việc loại bỏ cho đến khi chỉ còn các đỉnh không thể loại bỏ được nữa. Tập đỉnh này chính là một tập ổn định ngoài bé nhất của đồ thị G .

Ví dụ 3.5: Xét đồ thị có hướng sau:



Hình 3.3. Đồ thị và tập ổn định ngoài

$$T(a) = \{a\}; \quad T(d) = \{b, c, d\};$$

$$T(b) = \{a, b, c\}; \quad T(e) = \{e\};$$

$$T(c) = \{a, c\};$$

e là đỉnh cô lập phải giữ lại.

Loại bỏ a, c ta được tập $\{b, d, e\}$ là một tập ổn định ngoài bé nhất của G . Loại bỏ a, b ta được tập $\{c, d, e\}$ là một tập ổn định ngoài bé nhất khác của đồ thị G .

Giả sử cần phải xây dựng một hệ thống trạm bảo vệ cho tất cả các đối tượng trong một khu vực nào đó (nhà máy, kho hàng, trận địa...). Hiển nhiên, hệ thống trạm tối thiểu làm tròn được trách nhiệm chính là một tập ổn định ngoài bé nhất nào đó của đồ thị biểu diễn khu vực này.

3.3. NHÂN CỦA ĐỒ THỊ

Giả sử $G = (V, E)$ là một đồ thị.

Định nghĩa 3.3: Tập $B \subseteq V$ được gọi là *nhân* của đồ thị G nếu nó vừa là tập ổn định trong, vừa là tập ổn định ngoài của G , nghĩa là:

$$\forall x \in B : B \cap F(x) = \emptyset \text{ và}$$

$$\forall y \notin B : B \cap F(y) \neq \emptyset.$$

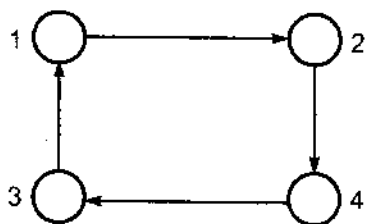
Hai điều kiện trên của nhân tương đương với đẳng thức:

$$F^{-1}(B) = V \setminus B.$$

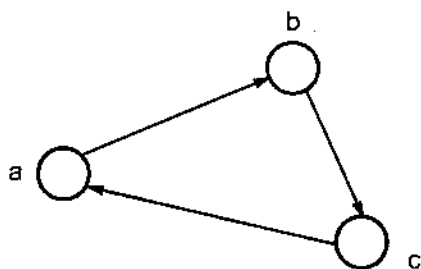
Từ định nghĩa của nhân, suy ra:

- Nhân không chứa đỉnh nút;
- Nếu $F(x) = \emptyset$ thì x phải thuộc vào một nhân nào đó của đồ thị.

Ví dụ 3.6: Xét các đồ thị sau đây:



Đồ thị có hai nhân



Đồ thị không có nhân

Hình 3.4. Đồ thị có nhân và đồ thị không có nhân

Chú ý: Nếu g là một hàm Grundy của đồ thị G thì tập hợp:

$$B = \{ x \mid g(x) = 0 \} \text{ là một nhân của } G.$$

Thật vậy, nếu x, y đều thuộc B thì $g(x) = g(y) (= 0)$ nên x không thể kề với y . Vậy B là tập ổn định trong.

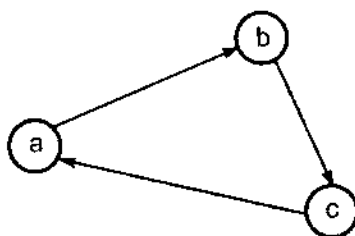
Mặt khác, nếu $x \notin B$ thì $g(x) > 0$. Khi đó, với $u = 0 < g(x)$ sẽ tồn tại $y \in F(x)$ sao cho $g(y) = u = 0$. Ta có $y \in B$. Vậy B là tập ổn định ngoài.

Định lý 3.2: Nếu B là nhân của đồ thị G thì B cũng là tập ổn định trong cực đại.

Chứng minh: Giả sử ngược lại, B không là tập ổn định trong cực đại. Điều này có nghĩa là tồn tại $a \notin B$ mà $B \cup \{a\}$ vẫn là tập ổn định trong. Vì B là nhân nên a sẽ kề với một đỉnh nào đó trong B . Vậy thì $B \cup \{a\}$ không thể là tập ổn định trong. Suy ra điều vô lý. Định lý được chứng minh. ■

Chú ý rằng, mệnh đề ngược lại là không đúng.

Ví dụ 3.7: Xét phản ví dụ sau:



Hình 3.5. Tập ổn định trong cực đại không phải là nhân

Tập $B = \{a\}$ là tập ổn định trong cực đại nhưng không phải là nhân của đồ thị.

Nhưng với đồ thị đối xứng, thì mệnh đề ngược lại của Định lý 3.2 là đúng.

Định lý 3.3: Trong đồ thị đối xứng không có đỉnh nút, mọi tập ổn định trong cực đại đều là nhân của đồ thị.

Chứng minh: Giả sử B là tập ổn định trong cực đại của đồ thị $G = (V, E)$. Ta chỉ cần chứng minh B là tập ổn định ngoài.

Thật vậy, giả sử $x \notin B$. Theo tính chất cực đại của B thì x phải kề với một đỉnh y nào đó trong B . Vì đồ thị G đối xứng nên $y \in F(x)$. Suy ra tập B là ổn định ngoài. Định lý được chứng minh. ■

Chú ý: Điều kiện G không có đỉnh nút là cần thiết, vì trong trường hợp ngược lại đỉnh x không nhất thiết phải kề với tập B .

Hệ quả 3.1: Mọi đồ thị đối xứng không có đỉnh nút luôn luôn có nhân.

Chứng minh: Chỉ cần tìm một tập ổn định trong cực đại. Mà tập ổn định trong cực đại thì luôn luôn có. ■

Định lý 3.4: Mọi đồ thị không có chu trình luôn có nhân.

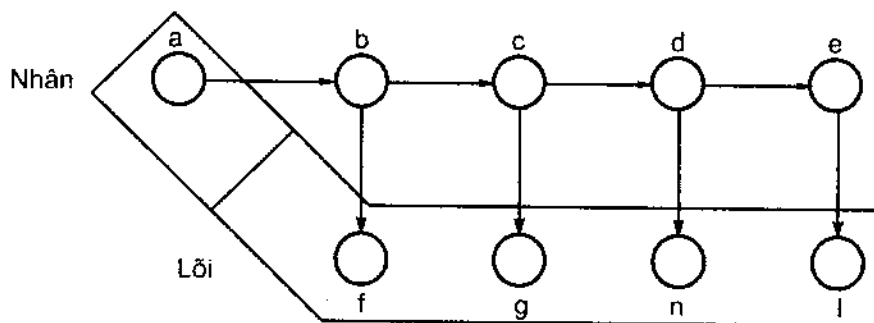
Chứng minh: Vì theo Định lý 2.1 đồ thị này có hàm Grundy, tập các đỉnh mà tại đó hàm Grundy bằng 0 chính là một nhân của đồ thị. ■

Vậy với điều kiện nào thì đồ thị có chu trình có nhân? Để trả lời câu hỏi này, cần đưa thêm vào khái niệm sau đây:

Định nghĩa 3.4: Tập con các đỉnh B được gọi là *lõi* của đồ thị $G = (V, E)$ nếu:

- 1) $\forall x, y \in B, x \neq y$: không tồn tại đường đi nối x với y ;
- 2) $\forall x \notin B$: có tồn tại đường đi từ x đến B .

Ví dụ 3.8: Lõi và nhân của một đồ thị:



Hình 3.6. Lõi và nhân của đồ thị

Trước hết ta có bổ đề sau:

Bổ đề 3.1: Mọi đồ thị đều có lõi.

Chứng minh: Chứng minh bằng quy nạp theo số đỉnh n của đồ thị G .

$n=1$: Đỉnh duy nhất cũng là lõi của đồ thị.

$(n) \Rightarrow (n+1)$: Đồ thị $G = (V, E)$ có $n+1$ đỉnh được xây dựng từ đồ thị $G_1 = (V_1, E_1)$ có n đỉnh thêm vào đỉnh a và một số cạnh kề a .
Thế thì, $V = V_1 \cup \{a\}$.

Theo giả thiết quy nạp, đồ thị G_1 có lõi là B_1 .

Nếu có đường đi từ a tới V_1 thì sẽ có đường từ a tới B_1 , do vậy B_1 cũng là lõi của G .

Ngược lại, giả sử không có đường đi từ a tới V_1 . Thế thì, không có cạnh đi ra từ a và a sẽ là đỉnh treo. Ký hiệu:

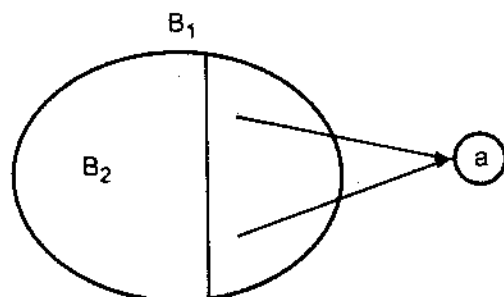
$$B_2 = \{x \mid x \in B_1 \text{ và không có đường đi từ } x \text{ tới } a\}.$$

Khi đó $B = B_2 \cup \{a\}$ là lõi của G . Thật vậy:

1) Giả sử $x, y \in B$ và $x \neq y$. Ta phải chứng minh không có đường nối x với y .

– Nếu x và y cùng thuộc B_2 thì chúng cùng thuộc B_1 . Mà B_1 là lõi của G_1 nên không có đường nối x với y trong G_1 . Hơn nữa, cũng không thể có đường nối qua đỉnh a vì a là đỉnh treo.

– Nếu $x = a, y \in B_2$ thì theo định nghĩa của B_2 sẽ không có đường đi từ y đến a , và cũng không có đường từ a đến y vì a là đỉnh treo.



Hình 3.7. Xây dựng lõi của đồ thị

2) Với $x \notin B$ thì $x \neq a$ và $x \notin B_2$.

Ta phải chứng tỏ có đường đi từ x đến B .

Giả sử $x \in B_1$. Vì $x \notin B_2$ nên có đường từ x đến a theo định nghĩa của B_2 .

Giả sử $x \notin B_1$. Vì $x \neq a$ nên $x \in V_1$. Suy ra có đường đi từ x đến $y \in B_1$ vì B_1 là lõi của G_1 . Nếu $y \notin B_2$ thì theo định nghĩa của B_2 sẽ có đường đi từ y đến a .

Trong tất cả các trường hợp đều suy ra có đường đi từ x đến B .

Bổ đề được chứng minh. ■

Định lý 3.5: Mọi đồ thị không có chu trình độ dài lẻ luôn có nhân.

Chứng minh: Phương hướng chứng minh như sau: Xây dựng ba dãy tập con các đỉnh của đồ thị: $V_0, V_1, V_2, \dots, B_0, B_1, B_2, \dots$ và $C_0, C_1, C_2, \dots$ lần lượt như sau:

Đặt: $V_0 = V$, chọn B_0 là lõi của V_0 và

$$C_0 = \{x \mid x \in V_0 \setminus B_0 \text{ và có cạnh đi từ } x \text{ đến } B_0\}.$$

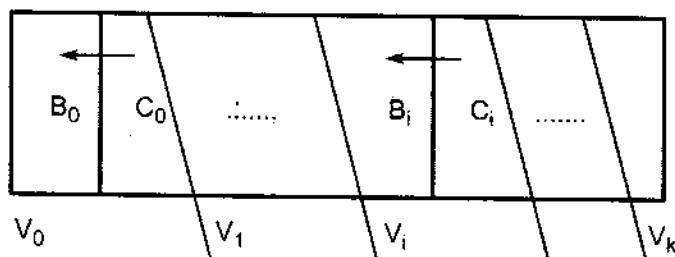
Lấy $V_1 = V_0 \setminus (B_0 \cup C_0)$

B_1 là lõi của V_1 và

$$C_1 = \{x \mid x \in V_1 \setminus B_1 \text{ và có cạnh đi từ } x \text{ đến } B_1\}.$$

Tương tự: $V_2 = V_1 \setminus (B_1 \cup C_1)$

.....



Hình 3.8. Cách xây dựng ba dãy tập con

Giả sử đã chọn được B_i là lõi của V_i .

Đặt $C_i = \{x \mid x \in V_i \setminus B_i \text{ và có cạnh đi từ } x \text{ đến } B_i\}$.

Đến một bước nào đó thì $V_k \setminus B_k = \emptyset$, khi đó đã vét hết các đỉnh của đồ thị.

Chọn tập $B = B_0 \cup B_1 \cup B_2 \cup \dots \cup B_k$. Ta chứng minh được tập B là nhân của đồ thị G . Phần chứng minh tiếp theo xin dành cho độc giả. ■

Định lý 3.6: Nếu mỗi đồ thị con của đồ thị G đều có nhân, thì G có hàm Grundy.

Chứng minh: Xây dựng hai dãy tập con các đỉnh: $V_0, V_1, V_2, \dots$ và $B_0, B_1, B_2, \dots$ lần lượt như sau:

$$V_0 = V.$$

Chọn B_0 là nhân của G .

$$V_1 = V_0 \setminus B_0$$

B_1 là nhân của đồ thị con tạo bởi V_1 .

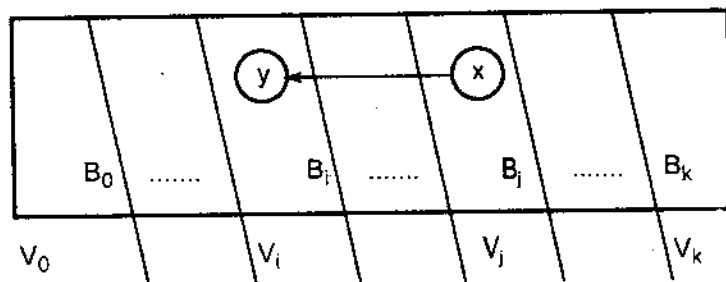
.....

$$V_{i+1} = V_i \setminus B_i$$

B_{i+1} là nhân của đồ thị con tạo bởi V_{i+1} .

Vì mỗi nhân đều khác rỗng, nên đến một bước nào đó sẽ vét hết các đỉnh của đồ thị và ta nhận được dãy các nhân $B_0, B_1, \dots, B_k$.

Xây dựng hàm g như sau: với $x \in B_i$, đặt $g(x) = i$. Ta chứng minh g là hàm Grundy của đồ thị.



Hình 3.9. Cách xây dựng dãy các nhân

1) Nếu x, y kề nhau thì chúng không thể thuộc cùng một tập B_i vì B_i là nhân, cho nên $g(x) \neq g(y)$.

2) Giả sử có số nguyên $i < g(x) = j$. Khi đó $x \in B_j \subseteq V_i \setminus B_i$. Vậy thì $x \notin B_i$.

Vì B_i là tập ổn định ngoài của V_i , mà $x \in V_i \setminus B_i$ nên tồn tại $y \in B_i$ sao cho $y \in F(x)$. Suy ra: $g(y) = i$.

Đó là điều phải chứng minh. ■

Hệ quả 3.2: Đồ thị đối xứng có hàm Grundy khi và chỉ khi nó không có đỉnh nút.

Chứng minh: Thật vậy, đồ thị có hàm Grundy thì không có đỉnh nút. Ngược lại, giả sử đồ thị G là đối xứng và không có đỉnh nút. Theo Hệ quả 3.1, mọi đồ thị con của G đều có nhân. Do vậy, đồ thị G có hàm Grundy. ■

Thuật toán 3.3 (Tìm nhân của đồ thị):

- 1) Chọn một tập ổn định ngoài bé nhất.
- 2) Kiểm tra xem nó có phải là tập ổn định trong hay không. Nếu đúng thì nhận được nhân bé nhất.
- 3) Tăng dần số phần tử của tập ổn định ngoài và lặp lại phép kiểm tra, để nhận được các nhân khác.

Chú ý: Nếu một đồ thị có số ổn định trong bé hơn số ổn định ngoài thì đồ thị ấy không có nhân.

3.4. ỨNG DỤNG NHÂN VÀO LÝ THUYẾT TRÒ CHƠI

Ta nghiên cứu loại trò chơi hai đấu thủ, được gọi là trò chơi Nim. Trò chơi này được mô tả như sau:

1. Có một tập hợp hữu hạn các *hình trạng* V .
2. Cho phép chuyển từ một hình trạng sang một số hình trạng khác, gọi là các *nước đi*.

3. Có một tập con các hình trạng được gọi là tập các *hình trạng kết thúc*.
4. Xuất phát từ một hình trạng, hai đấu thủ lần lượt chọn nước đi. Ai rơi vào hình trạng kết thúc là người thua cuộc.

Ta biểu diễn đồ thị $G = (V, F)$ cho trò chơi Nim như sau:

- Tập các đỉnh chính là tập các hình trạng V . Với mỗi hình trạng x thì $F(x)$ là tập các hình trạng có thể chuyển đến trực tiếp từ x bằng các nước đi.

- Nếu đồ thị biểu diễn trò chơi có nhân và tất cả các hình trạng kết thúc đều nằm trong nhân thì ta có chiến lược chắc chắn thắng sau đây:

- 1) Tìm cách đưa đối thủ vào nhân.
- 2) Khi đối thủ đã ở trong nhân thì đối thủ chọn nước đi nào cũng đều đi đến một hình trạng nằm ngoài nhân.
- 3) Đến lượt ta đi, khi đang ở ngoài nhân ta luôn chọn được nước đi để đưa đối thủ trở vào nhân. Đối thủ chắc chắn bị thua.

Ví dụ 3.9: Giả sử ta có m que và k là một số nguyên dương cho trước ($k \leq m$). Hai người tham gia cuộc chơi bốc que. Đến lượt đi, người chơi phải bốc một số que không vượt quá k . Ai bốc được chiếc que cuối cùng thì người đó thắng cuộc.

Hình trạng của trò chơi là số que có thể còn lại trên mặt đất. Đồ thị của trò chơi là $G = (V, F)$ với $V = \{m, m-1, \dots, 1, 0\}$ và $F(x) = \{x-1, x-2, \dots, x-k\}$ (trong tập này nếu có số âm thì bỏ đi). Để thấy rằng, hàm Grundy của đồ thị G sẽ là $g(x) = x \bmod (k+1)$ và nhân đồ thị là tập $B = \{x \mid x \bmod (k+1) = 0\}$. Số 0 là hình trạng kết thúc duy nhất cũng nằm trong nhân. Do vậy, nếu $m \notin B$ và ta được đi trước thì áp dụng chiến thuật trên, chắc chắn ta sẽ thắng.

Chẳng hạn, với $m = 10, k = 3$. Trò chơi được biểu diễn bằng đồ thị định hướng như ở Hình 3.10.

trò chơi này là một đồ thị định hướng có $m + 1$ đỉnh: $m, m - 1, \dots, 1, 0$ và cặp (i, j) là một cạnh khi và chỉ khi $i > j$. Hàm Grundy của đồ thị này là $g(x) = x$.

Đồ thị của trò chơi bốc ba đồng que ở trên là đồ thị tổng của ba trò chơi riêng biệt mà ta vừa xét, $G = G_1 + G_2 + G_3$.

Hàm Grundy của G là $g((x, y, z)) = g_1(x) \oplus g_2(y) \oplus g_3(z)$.

Nhân của đồ thị là $B = \{(x, y, z) \mid x \oplus y \oplus z = 0\}$. Hình trạng $(0, 0, 0)$ là hình trạng kết thúc duy nhất nằm trong nhân. Do vậy, có thể áp dụng chiến thuật chơi ở trên.

Chẳng hạn, với $m_1 = 6, m_2 = 5, m_3 = 2$.

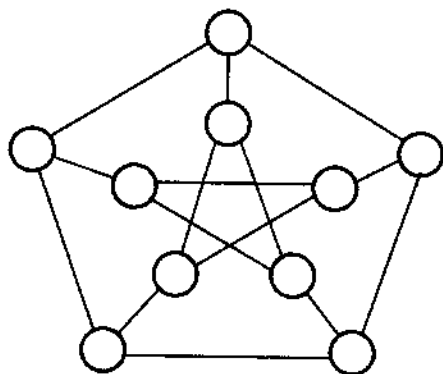
Nếu được đi trước, ta phải bốc ở đồng thứ hai 1 que để dẫn tới đỉnh $(6, 4, 2) \in B$. Sau đó đối thủ bốc thế nào cũng dẫn tới đỉnh nằm ngoài nhân, ta vẫn có thể bốc để dẫn đến đỉnh nằm trong nhân.

Chẳng hạn, anh ta bốc 3 que ở đồng thứ nhất thì ta bốc 3 que ở đồng thứ hai và dẫn tới hình trạng $(3, 1, 2) \in B$. Anh ta lại bốc 3 que ở đồng thứ nhất thì ta bốc 1 que ở đồng thứ ba để dẫn tới hình trạng $(0, 1, 1) \in B$. Anh ta bốc 1 que ở đồng nào thì ta bốc nốt que ở đồng còn lại và thắng cuộc.

Có thể mở rộng các trò chơi trên thành trò chơi với số đồng que tùy ý.

BÀI TẬP

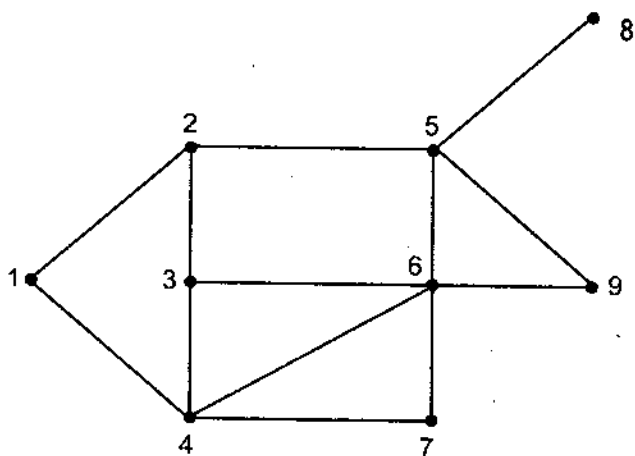
- 3.1. Tìm tập ổn định trong lớn nhất, tập ổn định ngoài bé nhất và nhân của đồ thị đầy đủ n đỉnh K_n .
- 3.2. Đồ thị chu trình C_n là đồ thị có n đỉnh $v_1, v_2, \dots, v_n$ và n cạnh $(v_1, v_2), (v_2, v_3), \dots, (v_n, v_1)$. Tìm số ổn định trong, số ổn định ngoài và nhân của đồ thị này.
- 3.3. Tìm tập ổn định trong lớn nhất, tập ổn định ngoài bé nhất của đồ thị Peterson sau:



Hình 3.11. Đồ thị Peterson

Đồ thị này có nhân hay không? Hãy tìm các lõi của đồ thị này.

- 3.4. Cho một đồ thị có 10 đỉnh, 15 cạnh và tìm lõi của nó.
- 3.5. Tìm tập ổn định trong lớn nhất, tập ổn định ngoài bé nhất, lõi và nhân của đồ thị được cho bởi hình sau:



Hình 3.12. Đồ thị vô hướng 9 đỉnh

Chương 4

CHU SỐ VÀ SẮC SỐ CỦA ĐỒ THỊ

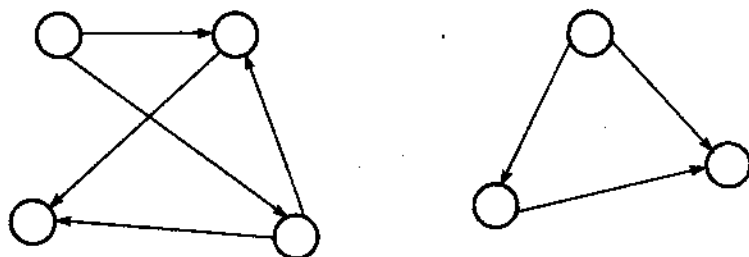
4.1. CHU SỐ CỦA ĐỒ THỊ

Cho đồ thị $G = (V, E)$ có n đỉnh, m cạnh, p thành phần liên thông.

Định nghĩa 4.1: Đại lượng $c = m - n + p$ được gọi là *chu số* của đồ thị G .

Trước hết, xét các tính chất của đại lượng này.

Ví dụ 4.1: Xét đồ thị sau:



Hình 4.1. Đồ thị định hướng không liên thông

Đồ thị trên có $n = 7$, $m = 8$ và $p = 2$. Vậy chu số $c = 8 - 7 + 2 = 3$.

Định lý 4.1: Nếu thêm một cạnh mới vào đồ thị G thì chu số tăng thêm 1 hoặc không thay đổi.

Chứng minh: Giả sử thêm cạnh mới (a, b) vào đồ thị G , khi đó m tăng thêm 1.

1) Nếu hai đỉnh a, b thuộc cùng một mảng liên thông trong G thì n, p không đổi, do vậy chu số tăng thêm 1.

2) Nếu hai đỉnh a, b nằm ở hai mảng liên thông khác nhau trong G thì p giảm 1, do vậy chu số không đổi. ■

Hệ quả 4.1: Chu số của đồ thị là số nguyên không âm.

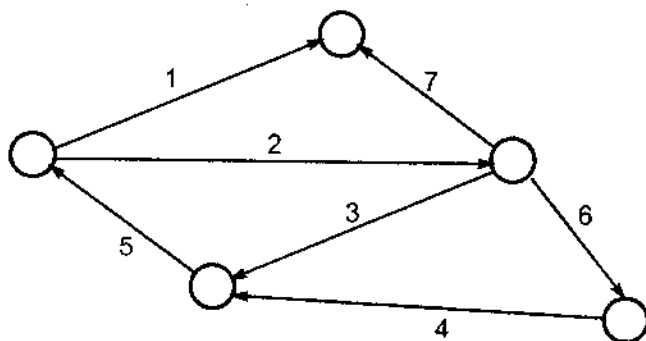
Chứng minh: Thật vậy, đồ thị G được xây dựng từ đồ thị G_0 gồm n đỉnh và không có cạnh nào cả. Sau đó, lần lượt thêm các cạnh vào đồ thị G_0 để được đồ thị G .

Chu số của G_0 là $c = 0 - n + n = 0$. Quá trình thêm cạnh không làm giảm chu số. Vậy chu số của $G \geq$ chu số của $G_0 = 0$. ■

Tìm ý nghĩa của chu số:

Đánh số các cạnh của đồ thị G theo một thứ tự nào đó: $1, 2, \dots, m$. Với mỗi chu trình vô hướng trong đồ thị G chọn một chiều thuận và biểu diễn nó bằng vectơ m chiều $(q_1, q_2, \dots, q_m)$ mà q_i là số lần xuất hiện của cạnh thứ i trong chu trình theo chiều thuận trừ đi số lần xuất hiện của cạnh đó trong chu trình theo chiều ngược.

Ví dụ 4.2: Xét đồ thị định hướng sau:



Hình 4.2. Đánh số các cạnh của đồ thị

Đồ thị có 7 cạnh, được đánh số như Hình 4.2, với chu trình vô hướng $[e_1, e_2, e_7]$ chọn chiều thuận là chiều e_1, e_2, e_7 , khi đó vectơ tương ứng sẽ là $(-1, 1, 0, 0, 0, 0, 1)$.

Do vậy, có thể đồng nhất mỗi chu trình vô hướng với một vectơ biểu diễn nó.

Các chu trình vô hướng $t_1, t_2, \dots, t_k$ được gọi là *độc lập tuyến tính* nếu các vectơ tương ứng với chúng lập thành một hệ độc lập tuyến tính.

Hệ chu trình đơn vô hướng $t_1, t_2, \dots, t_k$ được gọi là *độc lập tuyến tính cực đại* nếu nó là độc lập tuyến tính và mỗi chu trình đơn vô hướng của đồ thị đều có thể biểu diễn tuyến tính qua các chu trình của hệ.

Định lý 4.2: Chu số của đồ thị bằng số các chu trình đơn vô hướng độc lập cực đại trong đồ thị đó.

Chứng minh: Dùng phương pháp quy nạp theo số cạnh m của đồ thị.

– Nếu $m = 0$ thì chu số bằng 0, đồ thị không có chu trình đơn nào.

– $(m) \Rightarrow (m+1)$: Giả sử đồ thị G' có n đỉnh, $m + 1$ cạnh, p mảng liên thông. Có thể xem G' được xây dựng từ đồ thị G gồm m cạnh và bổ sung thêm một cạnh mới $e = (a, b)$. Đánh số cạnh e là cạnh thứ $m + 1$ của đồ thị G' .

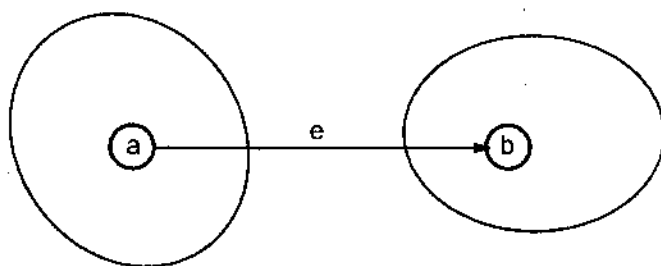
Theo giả thiết quy nạp, chu số của đồ thị G là $c(G) = m - n + p =$ số chu trình đơn vô hướng độc lập cực đại trong G . Ký hiệu các chu trình đó là $(T) = t_1, t_2, \dots, t_c$.

Hiển nhiên, mỗi chu trình trong G' không chứa e đều có thể biểu diễn tuyến tính qua hệ các chu trình (T) .

Ta xét hai trường hợp:

1) Hai đỉnh a, b của cạnh e nằm trong hai mảng liên thông khác nhau của G .

Vì số cạnh tăng 1, nhưng số mảng liên thông bị giảm 1, nên chu số của G' vẫn bằng chu số của G .



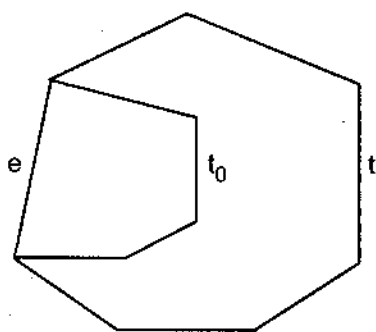
Hình 4.3. Hai mảng liên thông

Mặt khác, mỗi chu trình trong G' chứa e có tính chất sau: Số lần e xuất hiện trong chu trình theo chiều thuận bằng số lần e xuất hiện trong chu trình theo chiều ngược vì cạnh e là cầu nối duy nhất giữa hai mảng liên thông này của G . Do đó, thành phần thứ $m + 1$ của vector biểu diễn chu trình này bằng 0, và chu trình này vẫn có thể biểu diễn qua hệ (T) . Suy ra hệ (T) cũng chính là hệ chu trình đơn vô hướng độc lập cực đại của G' .

2) Hai đỉnh a, b của cạnh e thuộc cùng một mảng liên thông của G . Khi đó chu số $c(G') = c(G) + 1$. Chọn một đường đi đơn vô hướng trong G nối a với b rồi ghép thêm cạnh e ta được một chu trình đơn vô hướng trong G' . Ký hiệu chu trình này là t_0 .

Xét hệ $(T') = t_0, (T) = t_0, t_1, t_2, \dots, t_c$ gồm $c(G) + 1$ chu trình đơn vô hướng trong G' .

Hệ (T') là độc lập tuyến tính, vì (T) độc lập tuyến tính và t_0 không thể biểu diễn được qua (T) vì toạ độ thứ $m + 1$ của vector biểu diễn t_0 bằng 1, còn của các vector biểu diễn các chu trình trong (T) thì bằng 0.



Hình 4.4. Hai chu trình chung một cạnh

Giả sử t là một chu trình nào đó của G' chứa e . Chọn chiều của t sao cho chu trình tổng $t + t_0$ không chứa e . Vậy thì chu trình tổng $t + t_0$ có thể biểu diễn tuyến tính qua hệ (T) . Do đó, chu trình t cũng có thể biểu diễn tuyến tính qua hệ (T) . Vậy (T') là hệ chu trình đơn vô hướng độc lập cực đại của G' . ■

Đồ thị có chu số bằng 0 được gọi là đồ thị *phi chu trình*. Lớp đồ thị phi chu trình là lớp đặc biệt, nhưng hay gặp trong thực tế ứng dụng. Trước hết chỉ ra một đặc trưng của lớp đồ thị này như sau:

Định lý 4.3: Đồ thị định hướng $G = (V, E)$ là phi chu trình khi và chỉ khi các đỉnh của nó luôn có thể đánh số để sao cho mỗi cạnh (i, j) của đồ thị đều thoả mãn $i < j$.

Chứng minh:

a) Nếu có thể đánh số các đỉnh như trên thì hiển nhiên đồ thị không có chu trình.

b) Để chứng minh điều ngược lại, xây dựng thuật toán sau để đánh số các đỉnh của đồ thị định hướng phi chu trình:

Thuật toán dựa trên một tính chất rất đơn giản: Trong một đồ thị định hướng, không rỗng, phi chu trình tùy ý luôn tồn tại đỉnh mà không có một cạnh nào đi vào đỉnh đó. Trước hết, thuật toán tính bậc vào cho các đỉnh của đồ thị.

– Những đỉnh có bậc vào bằng 0 sẽ được đưa vào stack (ngăn xếp – LIFO). Đánh số cho đỉnh đang ở đỉnh stack, loại bỏ đỉnh này khỏi stack và giảm bậc vào cho các đỉnh kề với đỉnh này. Nếu có đỉnh mà bậc vào đã giảm hết thì nạp nó lên đỉnh của stack.

– Tiếp tục quá trình đánh số tăng dần, loại đỉnh, giảm bậc vào... cho đến khi stack trở thành rỗng. Và ta đã đánh số xong tất cả các đỉnh của đồ thị. ■

Dựa vào chứng minh của định lý trên, xây dựng thuật toán đánh số các đỉnh cho đồ thị định hướng phi chu trình như sau:

Thuật toán 4.1 (Đánh số các đỉnh của đồ thị phi chu trình):

Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị phi chu trình G .

Kết quả: Mảng SO các số nguyên với $SO[v]$ là số đánh trên đỉnh v .

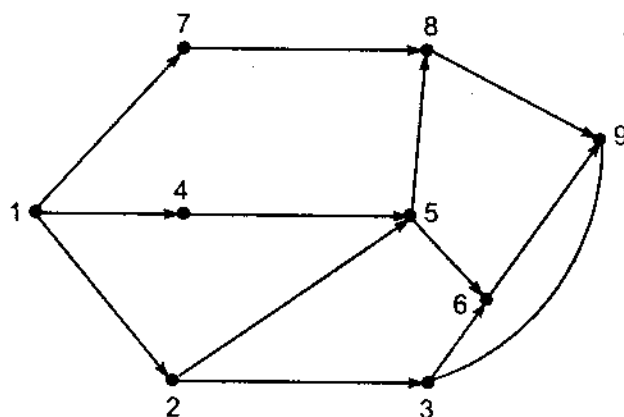
```

1  Begin
2  for  $v \in V$  do  $BAC\_V[v] := 0$  ;
      {  $BAC\_V[v]$  chứa bậc vào của đỉnh  $v$  }
3  for  $u \in V$  do
4      for  $v \in DK[u]$  do  $BAC\_V[v] := BAC\_V[v] + 1$  ;
5   $S := \emptyset$  ;
6  for  $v \in V$  do
7      if  $BAC\_V[v] = 0$  then push  $v$  onto  $S$  ;
8   $k := 0$  ;
9  while  $S \neq \emptyset$  do
10     begin  $u := \text{top}(S)$  ;  $\text{pop}(S)$  ;
11          $k := k + 1$  ;  $SO[u] := k$  ;
12         for  $v \in DK[u]$  do
13             begin  $BAC\_V[v] := BAC\_V[v] - 1$  ;
14                 if  $BAC\_V[v] = 0$  then push  $v$  onto  $S$ 
15             end
16     end
17  End.

```

Độ phức tạp của thuật toán là $O(m + n)$.

Ví dụ 4.3: Áp dụng thuật toán trên để đánh số các đỉnh cho đồ thị phi chu trình sau:



Hình 4.5. Các đỉnh của đồ thị phi chu trình đã được đánh số

Việc đánh số các đỉnh trên đồ thị định hướng phi chu trình có nhiều ứng dụng trong sơ đồ PERT, phương pháp đường tới hạn CPM...

4.2. SẮC SỐ CỦA ĐỒ THỊ

Khái niệm sắc số liên quan đến bài toán tô màu đồ thị như sau: Hãy tô màu các đỉnh của một đồ thị đã cho, sao cho hai đỉnh kề nhau phải được tô bằng hai màu khác nhau.

Ta nói rằng, đồ thị G tô được bằng k màu nếu tồn tại hàm:

$$m : V \rightarrow \{0, 1, 2, \dots, k-1\}$$

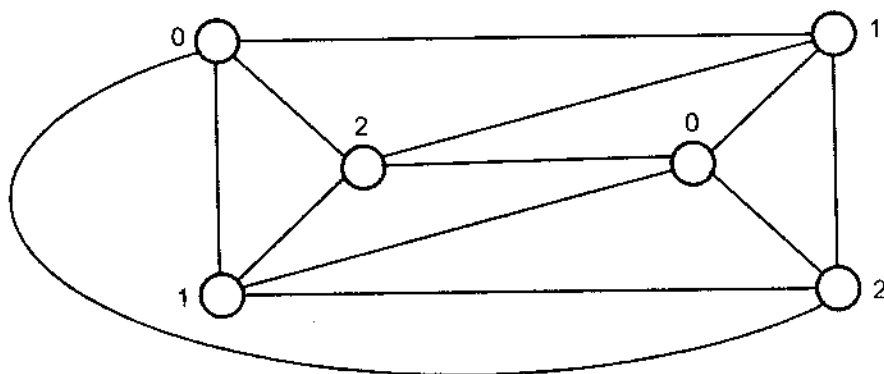
sao cho nếu hai đỉnh x và y kề nhau thì $m(x) \neq m(y)$.

Dễ thấy rằng, đồ thị G tô màu được khi và chỉ khi nó không có đỉnh nút.

Định nghĩa 4.2: Sắc số của một đồ thị chính là số màu ít nhất dùng để tô các đỉnh của đồ thị đó.

Ta ký hiệu số s là sắc số của đồ thị G . Hiển nhiên $s \leq n$, số màu không vượt quá số đỉnh của đồ thị.

Ví dụ 4.4: Hãy tô màu đồ thị sau:



Hình 4.6. Tô màu các đỉnh đồ thị

Đồ thị trên có sắc số bằng 3.

Nhận xét: Mỗi cách tô màu m cho đồ thị G sẽ ứng với một cách phân hoạch tập đỉnh V thành các tập ổn định trong không giao nhau, mỗi tập ứng với một màu. Ngược lại, mỗi cách phân hoạch tập đỉnh V thành các tập ổn định trong không giao nhau sẽ cho ta một cách tô màu.

Định lý 4.4: Mọi chu trình độ dài lẻ luôn có sắc số bằng 3.

Chứng minh: Giả sử chu trình có độ dài là $2n + 1$.

Chứng minh bằng phương pháp quy nạp theo n .

$n = 1$: Chu trình gồm ba đỉnh, mà hai đỉnh bất kỳ đều kề nhau. Vậy phải dùng đúng ba màu để tô các đỉnh.

$(n) \Rightarrow (n + 1)$: Giả sử α là một chu trình có độ dài $2(n + 1) + 1 = 2n + 3$ với dãy các đỉnh là $[x_1, x_2, \dots, x_{2n+1}, x_{2n+2}, x_{2n+3}]$.

Nối x_1 với x_{2n+1} được một chu trình α' có độ dài $2n + 1$. Theo giả thiết quy nạp, chu trình α' có sắc số bằng 3. Lấy màu của x_1 tô cho x_{2n+2} , còn màu của x_{2n+1} tô cho x_{2n+3} . Chu trình α đã được tô màu mà không phải thêm màu mới. Vậy chu trình α có sắc số bằng 3. ■

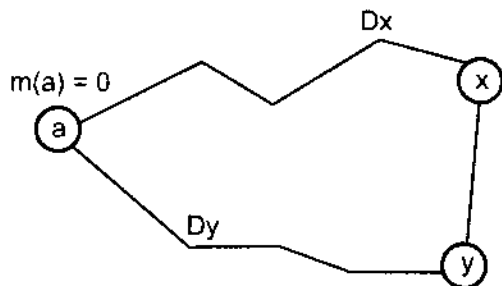
Định lý 4.5: Đồ thị đầy đủ n đỉnh K_n có sắc số bằng n .

Dưới đây là một tiêu chuẩn đơn giản để kiểm tra xem một đồ thị có hai sắc (sắc số bằng 2) hay không.

Định lý 4.6 (Konig): Giả sử đồ thị G có ít nhất một cạnh. Đồ thị G là hai sắc khi và chỉ khi G không có chu trình đơn vô hướng độ dài lẻ.

Chứng minh: Giả sử G là đồ thị hai sắc. Theo Định lý 4.4 thì G không thể có chu trình đơn vô hướng độ dài lẻ.

Ngược lại, giả sử G không có chu trình đơn vô hướng độ dài lẻ. Không mất tính tổng quát, có thể xem G là liên thông. Chọn một đỉnh a nào đó trong đồ thị.



Hình 4.7. Cách xây dựng hàm tô màu

Đặt $m(a) = 0$.

Với $x \neq a$ ta ký hiệu $d(x)$ là độ dài đường đi vô hướng ngắn nhất nối a với x .

Đặt $m(x) = d(x) \bmod 2$, sẽ chứng minh m là hàm màu của G .

Giả sử x, y kề nhau, lấy Dx là đường đi vô hướng ngắn nhất nối a với x có độ dài $d(x)$, và Dy là đường đi vô hướng ngắn nhất nối a với y có độ dài $d(y)$. Chu trình đơn $[Dx, (x, y), Dy]$ có độ dài $d(x) + d(y) + 1$ phải là một số chẵn.

Vậy thì $d(x) + d(y)$ là một số lẻ, có nghĩa là $d(x)$ và $d(y)$ khác nhau tính chẵn lẻ.

Do vậy: $m(x) \neq m(y)$.

Hàm tô màu m có hai giá trị, vậy sắc số ≤ 2 . G có ít nhất một cạnh nên sắc số của nó bằng 2. ■

Từ định lý trên ta có hệ quả sau:

Hệ quả 4.2: Tất cả các chu trình độ dài chẵn đều có sắc số bằng 2.

Kết quả dưới đây cho một thuật toán tốt để tìm sắc số của một đồ thị vô hướng.

Định lý 4.7: Đồ thị vô hướng G có sắc số bằng s khi và chỉ khi G có hàm Grundy $g \leq s - 1$.

Chứng minh:

$\Leftarrow$) Nếu đồ thị G có hàm Grundy $g \leq s - 1$ thì chỉ việc chọn g làm hàm tô màu.

$\Rightarrow$) Ngược lại, giả sử đồ thị G có sắc số là s , nghĩa là tồn tại hàm tô màu m với tập màu là $\{0, 1, \dots, s - 1\}$. Đồ thị G không có đỉnh nút.

Hàm tô màu m sẽ phân hoạch tập đỉnh V thành các tập ổn định trong không rỗng, không giao nhau:

$$C_i = \{x \mid m(x) = i\}, i = 0, 1, \dots, s - 1.$$

Với mỗi tập ổn định trong của đồ thị vô hướng luôn có thể bổ sung các đỉnh để thành cực đại, và đó cũng là nhân của đồ thị.

Xây dựng hai dãy tập con các đỉnh $V_0, V_1, V_2, \dots$ và $B_0, B_1, B_2, \dots$ lần lượt như sau: $V_0 = V$.

Vì C_0 là tập ổn định trong của V_0 nên có thể bổ sung để thành tập B_0 là nhân của V_0 . Hiển nhiên $B_0 \subseteq V_0$.

$$V_1 = V_0 \setminus B_0$$

Vì $C_1 \setminus B_0$ là tập ổn định trong của V_1 nên có thể bổ sung để thành tập B_1 là nhân của V_1 . Ta có:

$$C_1 \setminus B_0 \subseteq B_1 \subseteq V_1$$

.....

$$V_{i+1} = V_i \setminus B_i$$

Vì $C_{i+1} \setminus (B_0 \cup \dots \cup B_i)$ là tập ổn định trong của V_{i+1} nên có thể bổ sung để thành tập B_{i+1} là nhân của V_{i+1} . Ta có

$$C_{i+1} \setminus (B_0 \cup \dots \cup B_i) \subseteq B_{i+1} \subseteq V_{i+1}$$

Quá trình tiếp tục cho đến V_{k-1} .

$$C_{k-1} \setminus (B_0 \cup \dots \cup B_{k-2}) \text{ là tập ổn định trong của } V_{k-1}.$$

Sau khi bổ sung thành nhân B_{k-1} ta có

$$C_{k-1} \setminus (B_0 \cup \dots \cup B_{k-2}) \subseteq B_{k-1} \subseteq V_{k-1}.$$

Ta có

$$C_{k-1} = (C_{k-1} \cap (B_0 \cup \dots \cup B_{k-2})) \cup (C_{k-1} \setminus (B_0 \cup \dots \cup B_{k-2}))$$

$$\subseteq (B_0 \cup \dots \cup B_{k-2}) \cup B_{k-1} = B_0 \cup \dots \cup B_{k-1}$$

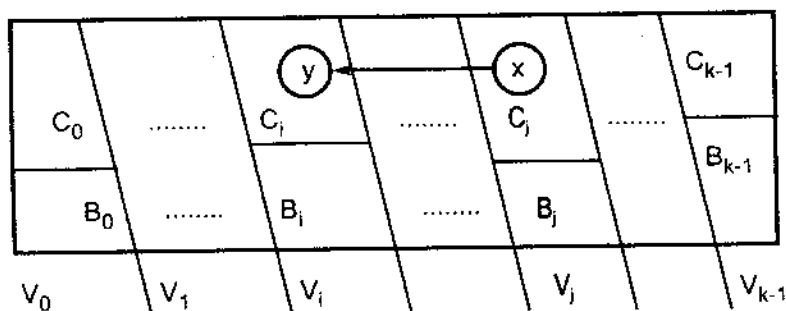
$$V = C_0 \cup \dots \cup C_{k-1} \subseteq B_0 \cup \dots \cup B_{k-1}$$

Vậy đến nhân B_{k-1} thì đã vét hết các đỉnh của V .

Ta được dãy: $B_0, B_1, \dots, B_{k-1}$, trong đó B_i là nhân của V_i .

Bây giờ xây dựng hàm Grundy cho đồ thị G .

Với $x \in B_i$ đặt $g(x) = i$ và chứng minh g chính là một hàm Grundy của đồ thị G .



Hình 4.8. Cách xây dựng dãy các nhân

1) Nếu x, y kề nhau thì không thể cùng nằm trong một tập B_i vì B_i là nhân, cho nên $g(x) \neq g(y)$.

2) Giả sử có $u < g(x) = j$.

Khi đó $x \notin B_u$. Vì B_u là tập ổn định ngoài của V_u nên tồn tại $y \in B_u$ sao cho $y \in F(x)$. Suy ra $g(y) = u$, đó là điều phải chứng minh. ■

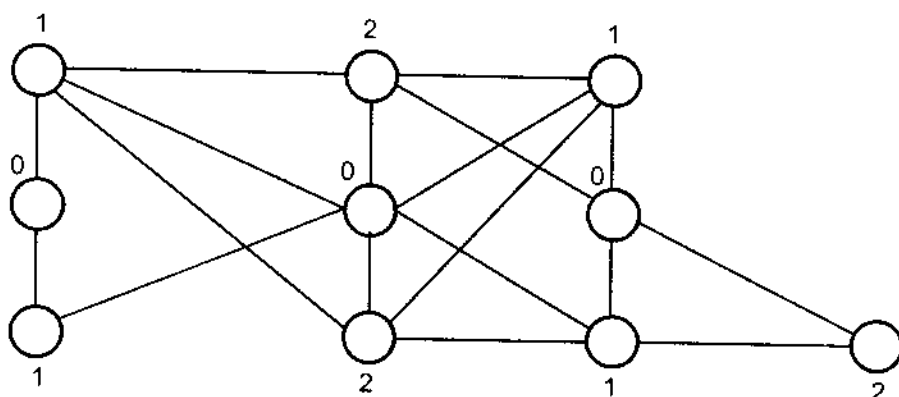
Hệ quả 4.3: Mọi đồ thị vô hướng không có đỉnh nút đều có hàm Grundy và giá trị cực đại của các hàm này phải bằng nhau và bằng sắc số của đồ thị trừ đi 1.

Thuật toán 4.2 (Tô màu đồ thị không có đỉnh nút):

- 1) Liệt kê các đỉnh $x_1, x_2, \dots, x_n$ của đồ thị theo thứ tự giảm dần của bậc: $r(x_1) \geq r(x_2) \geq \dots \geq r(x_n)$ để làm giảm các phép kiểm tra ở bước dưới.
- 2) Tô màu 0 cho đỉnh x_1 (đỉnh có bậc lớn nhất) cùng các đỉnh không kề với x_1 và không kề với các đỉnh đã tô màu 0.
- 3) Lặp lại thủ tục tô màu $i + 1$ giống như thủ tục tô màu i cho đến khi tô màu hết các đỉnh của đồ thị.

Số màu đã dùng chính là sắc số của đồ thị.

Ví dụ 4.5: Tô màu đồ thị sau đây:



Hình 4.9. Tô màu một đồ thị

Định lý 4.8: Giả sử đồ thị G tô được bằng $s + 1$ màu, đồ thị H tô được bằng $t + 1$ màu. Khi đó đồ thị tổng $G + H$ tô được bằng $d + 1$ màu, trong đó:

$$d = \max \{ s' \oplus t' \mid s' \leq s, t' \leq t \}.$$

Chứng minh: Theo Định lý 4.7 đồ thị G có hàm Grundy $g \leq s$, đồ thị H có hàm Grundy $h \leq t$. Ta có $z((x, y)) = g(x) \oplus h(y)$ là hàm Grundy của đồ thị tổng $G + H$. Giá trị lớn nhất của hàm z là d . Từ đó suy ra kết quả. ■

Ví dụ 4.6: Đồ thị G tô được bằng 7 màu, đồ thị H tô được bằng 5 màu thì đồ thị tổng $G + H$ tô được bằng 8 màu.

Định lý 4.9: Nếu đồ thị G có n đỉnh và sắc số s thì số ổn định trong u của đồ thị G sẽ không nhỏ hơn $\frac{n}{s}$.

Chứng minh: Lập các tập đỉnh cùng màu:

$C_i = \{x \mid x \text{ tô màu } i\}, i = 0, \dots, s - 1$ là các tập ổn định trong.

$$|C_i| \leq u$$

$$n = \sum_{i=0}^{s-1} |C_i| \leq s \cdot u. \text{ Suy ra: } u \geq \frac{n}{s}. \blacksquare$$

Định lý 4.10: Nếu bậc lớn nhất của các đỉnh trong đồ thị G là r thì sắc số của đồ thị $G \leq r + 1$.

Chứng minh: Chứng minh bằng quy nạp theo số đỉnh n .

$n = 1$: Bậc của đỉnh bằng 0 và sắc số bằng 1.

$n = 2$: Bậc của các đỉnh bằng 0 thì sắc số bằng 1, còn bậc của các đỉnh bằng 1 thì sắc số bằng 2.

$(n) \Rightarrow (n + 1)$: Giả sử đồ thị G có n đỉnh, các đỉnh có bậc cao nhất là r . Theo giả thiết quy nạp: $s(G) \leq r + 1$.

Đồ thị G' có $n + 1$ đỉnh được xem là đồ thị G có n đỉnh và thêm một đỉnh mới a và một số cạnh kề nó. Khi đó: $s(G) \leq s(G') \leq s(G) + 1$.

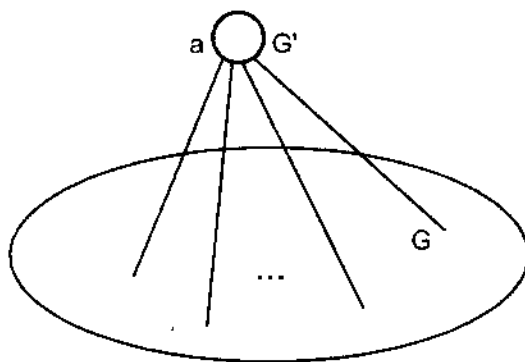
Giả sử bậc cao nhất của các đỉnh trong G' là r' . Hiển nhiên: $r \leq r'$.

Nếu $s(G) \leq r$ thì: $s(G') \leq r + 1 \leq r' + 1$.

Nếu $s(G) = r + 1$ thì:

1) Trường hợp: $r < r'$ ta có:

$$s(G') \leq r + 1 + 1 \leq r' + 1$$

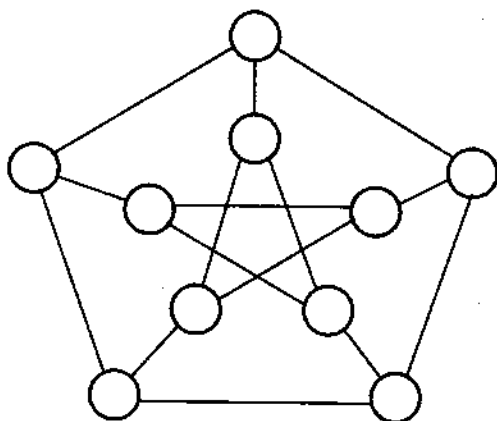


Hình 4.10. Cách chọn màu cho đỉnh mới

2) Trường hợp: $r = r'$ thì đỉnh mới a được nối với không quá r đỉnh, do vậy chỉ cần giữ nguyên cách tô màu của G thì các đỉnh kề với a được tô không quá r màu và vẫn còn thừa một màu dành cho đỉnh a , suy ra: $s(G') = s(G) \leq r + 1 = r' + 1$. ■

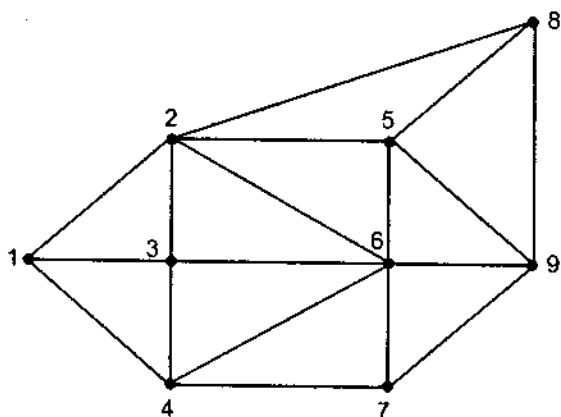
BÀI TẬP

- 4.1. Đồ thị đầy đủ n đỉnh K_n có chu số và sắc số bằng bao nhiêu?
- 4.2. Hãy tô màu cho đồ thị Peterson sau:



Hình 4.11. Đồ thị Peterson 10 đỉnh

- 4.3. Chọn một số đồ thị. Tô màu và tìm sắc số của các đồ thị này.
- 4.4. Giả sử $G = (V, E)$ là một đồ thị có n đỉnh. Đồ thị $G = (V, V \times V \setminus E)$ được gọi là *đồ thị bù* của G . Hãy chứng minh rằng:
- 1) $s(G) + s(\overline{G}) \leq n + 1$.
 - 2) $s(G) \cdot s(\overline{G}) \geq n$.
- 4.5. Tô màu và tìm sắc số của đồ thị được cho bởi hình sau:



Hình 4.12. Đồ thị vô hướng 9 đỉnh

Chương 5

CẶP GHÉP VÀ ĐỒ THỊ HAI PHẦN

5.1. TẬP ĐỈNH TỤ VÀ CẶP GHÉP

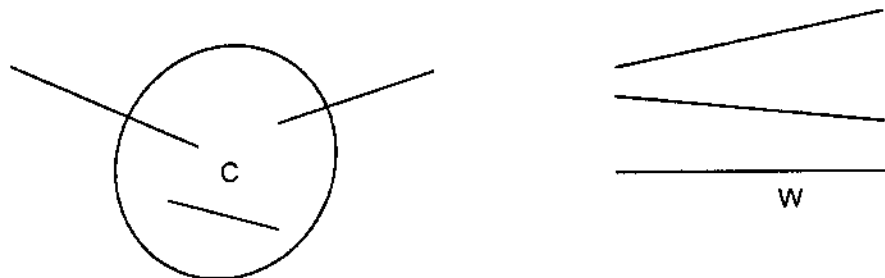
Để đưa vào các khái niệm mới này, xét bài toán phân công nhiệm vụ như sau:

Một cơ quan có n nhân viên $x_1, x_2, \dots, x_n$ và m nhiệm vụ $y_1, y_2, \dots, y_m$. Do quá trình đào tạo, mỗi nhân viên có thể đảm nhiệm một hay nhiều nhiệm vụ, mỗi nhiệm vụ có một số nhân viên có thể đảm nhiệm được. Vậy có thể phân công cho mỗi nhân viên đảm nhiệm một nhiệm vụ thích hợp với trình độ của người đó được không?

Bài toán này sẽ giải quyết được nhờ khái niệm cặp ghép:

Định nghĩa 5.1: Giả sử G là một đồ thị vô hướng.

- 1) Tập con các đỉnh C của G được gọi là *tập đỉnh tụ* của đồ thị G nếu mỗi cạnh của G đều kề với ít nhất một đỉnh nào đó trong C .
- 2) Tập con các cạnh W của G được gọi là *cặp ghép* nếu trong W không có hai cạnh nào kề nhau.



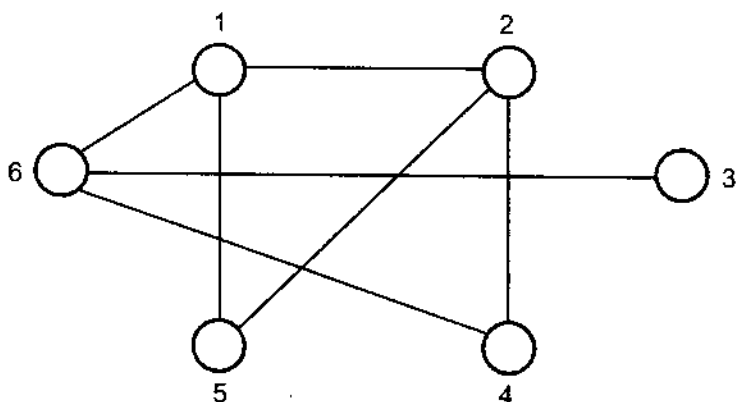
Hình 5.1. Tập đỉnh tụ và cặp ghép

Tập đỉnh tựa của một đồ thị luôn tồn tại. Tập tất cả các đỉnh là một ví dụ về tập đỉnh tựa của đồ thị. Song thường quan tâm đến tập đỉnh tựa có ít đỉnh nhất. Dễ thấy, tập con các đỉnh C là tập đỉnh tựa khi và chỉ khi tập $V \setminus C$ là ổn định trong.

Cặp ghép của một đồ thị cũng luôn tồn tại. Mỗi cạnh trong cặp ghép tạo nên sự ghép giữa một đỉnh với đỉnh kề của nó.

Chúng ta cũng thường quan tâm đến các cặp ghép có nhiều cạnh nhất.

Ví dụ 5.1: Xét đồ thị vô hướng cho ở Hình 5.2 sau:



Hình 5.2. Đồ thị vô hướng

Đồ thị này có các tập đỉnh tựa là: $\{1, 2, 6\}$, $\{2, 5, 6\}$, ... và các cặp ghép: $\{(1, 2), (3, 6)\}$, $\{(1, 5), (2, 4), (3, 6)\}$, ...

5.2. ĐỒ THỊ HAI PHẦN

Xét lớp đồ thị đặc biệt sau:

Định nghĩa 5.2: Đồ thị $G = (V, F)$ được gọi là đồ thị hai phần nếu tập đỉnh V có thể tách thành hai tập ổn định trong không giao nhau.

Hay nói một cách khác:

$$V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset,$$

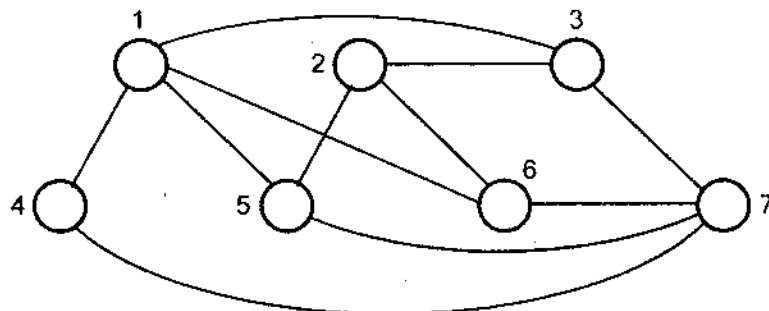
$$F(V_1) \subseteq V_2, F(V_2) \subseteq V_1.$$

Khi đó mỗi cạnh có một đầu thuộc V_1 và đầu kia thuộc V_2 . Đồng thời, V_1 và V_2 là các tập đỉnh tựa của đồ thị G .

Nếu đồ thị có ít nhất một cạnh, thì khái niệm đồ thị hai phần trùng với điều kiện sắc số bằng 2.

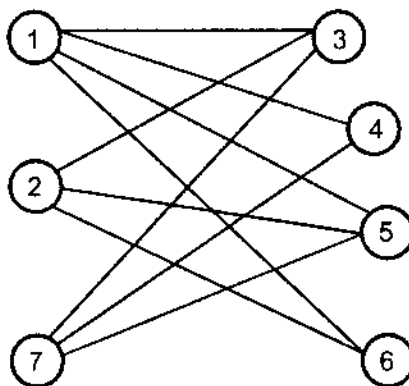
Đồ thị hai phần thường ký hiệu là: $G = (V_1, V_2, F)$.

Ví dụ 5.2: Cho đồ thị vô hướng:



Hình 5.3. Đồ thị vô hướng

Vẽ lại đồ thị này ta nhận được:



Hình 5.4. Đồ thị hai phần tương ứng

Đồ thị trên là đồ thị hai phần có tập đỉnh tựa bé nhất là $\{1, 2, 7\}$ và cặp ghép lớn nhất là $\{(1, 3), (2, 5), (4, 7)\}$.

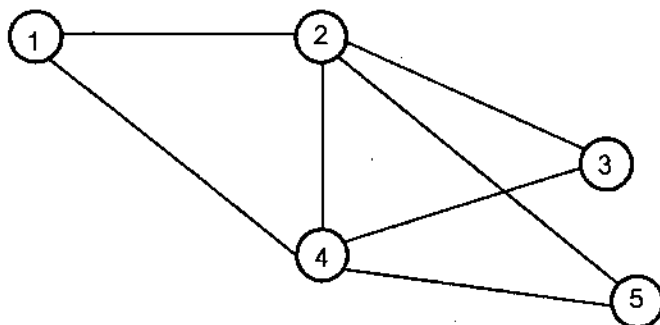
Đồ thị trong Ví dụ 5.1 không phải là đồ thị hai phần.

Để kiểm tra xem một đồ thị vô hướng G có phải là đồ thị hai phần hay không, sử dụng thuật toán sau đây:

Thuật toán 5.1 (Kiểm tra một đồ thị là đồ thị hai phần):

- 1) Chọn một đỉnh bất kỳ a trong đồ thị G .
- 2) Đặt $V_1 = \{a\}$.
- 3) Đặt V_2 là tập các đỉnh kề của đỉnh trong V_1 .
- 4) Nếu $V_1 \cap V_2 \neq \emptyset$ thì kết luận đồ thị không phải là đồ thị hai phần, ngược lại thì quay lại bước 3) cho đến khi hết đỉnh để thêm vào.
- 5) Cuối cùng, nếu $V_1 \cap V_2 = \emptyset$ thì kết luận đồ thị là đồ thị hai phần.

Ví dụ 5.3: Xét đồ thị vô hướng:



Hình 5.5. Đồ thị vô hướng

Bắt đầu chọn: $V_1 = \{1\}$, $V_2 = \{2, 4\}$.

Sau đó thêm vào $V_1 = \{1, 2, 3, 4, 5\}$, ta có: $V_1 \cap V_2 \neq \emptyset$.

Vậy đồ thị trên không là đồ thị hai phần.

Nếu bỏ cạnh $(2, 4)$ thì đồ thị trên trở thành đồ thị hai phần.

Giả sử G là đồ thị hai phần có n đỉnh. Ký hiệu k là số phần tử của tập đỉnh tựa bé nhất. Ta có định lý sau:

Định lý 5.1:

1. Số ổn định trong của đồ thị hai phần G bằng $n - k$.
2. Số phần tử của cặp ghép lớn nhất của G bằng k .

Chứng minh:

1. Từ nhận xét: C là tập đỉnh tựa nhỏ nhất $\Leftrightarrow V \setminus C$ là tập ổn định trong lớn nhất.

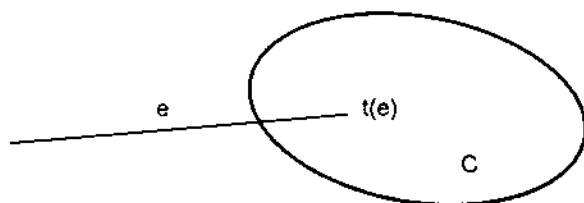
2. Giả sử W là cặp ghép lớn nhất và C là tập tựa nhỏ nhất.

Lập ánh xạ $t: W \rightarrow C$ như sau:

$$\forall e \in W, t(e) \text{ là một đỉnh của } e \text{ thuộc } C.$$

Đỉnh đó tồn tại vì C là tập tựa, còn nếu cả hai đỉnh của e đều thuộc C thì lấy một trong hai đỉnh đó.

Nếu $u, v \in W$ và $u \neq v$ thì $t(u) \neq t(v)$ vì hai cạnh u và v không có đỉnh chung. Vậy thì: $|W| \leq |C| = k$.



Hình 5.6. Cách xây dựng ánh xạ t

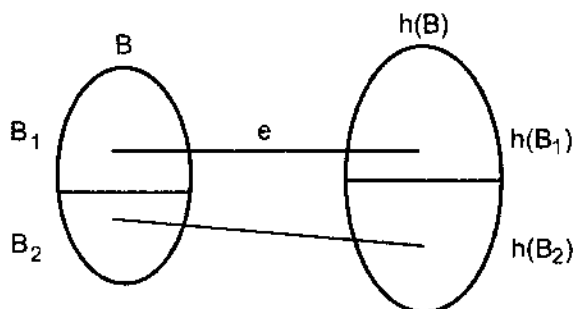
Để chứng minh chiều ngược lại, xây dựng tập đỉnh tựa C từ cặp ghép lớn nhất W mà hai tập này có cùng lực lượng.

Ký hiệu B là tập các đỉnh của W trong V_1 . Lập ánh xạ $h: B \rightarrow V_2$ như sau:

$$\forall a \in B, \exists b \in V_2 : (a, b) \in W \text{ ta đặt } h(a) = b.$$

Vậy $h(B)$ chính là tập các đỉnh của W trong V_2 .

Nếu $a, c \in B$ và $a \neq c$ thì $h(a) \neq h(c)$ vì các cạnh trong W chứa a và c không kề nhau.



Hình 5.7. Cách xây dựng tập đỉnh tựa

Một đường đi trong đồ thị G được gọi là *đường đơn* nếu nó có dạng

$$\langle w_1 u_1 w_2 u_2 \dots w_q u_q \rangle$$

trong đó: $w_1, w_2, \dots, w_q$ đều thuộc W , còn $u_1, u_2, \dots, u_q$ đều không thuộc W .

Ký hiệu: $B_1 = \{a \in B \mid \exists \text{ đường đơn đi từ } a \text{ đến một đỉnh nào đó nằm ngoài } B\}$ và $B_2 = B \setminus B_1$.

Đặt: $C = B_2 \cup h(B_1)$.

Chúng ta sẽ chứng minh rằng, tập C là tập đỉnh tựa của đồ thị G . Chứng minh điều này bằng phản chứng.

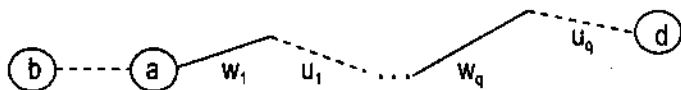
Giả sử rằng, tập C không phải tập đỉnh tựa của đồ thị hai phần G , nghĩa là có cạnh (a, b) nào đó không tựa vào tập C . Vậy thì $a \notin B_2$ và $b \notin h(B_1)$.

Nhưng vì tập các cạnh W là cặp ghép lớn nhất, nên cạnh (a, b) phải kề với một cạnh nào đó trong W , nghĩa là: $a \in B$ hoặc $b \in h(B)$.

Xét các trường hợp sau đây:

1) Trường hợp: $a \in B$, suy ra: $a \in B_1$.

Khi đó có một đường đơn $(X) = \langle w_1 u_1 w_2 u_2 \dots w_q u_q \rangle$ dẫn đỉnh a tới một đỉnh d nào đó nằm ngoài tập B .



– Nếu $b \notin h(B)$ thì cạnh $(a, b) \notin W$, loại các cạnh $w_1, w_2, \dots, w_q$ ra khỏi W và thay các cạnh $(a, b), u_1, u_2, \dots, u_q$ vào W . Khi đó, W vẫn là một cặp ghép và số cạnh tăng thêm 1. Vậy trái với giả thiết W là cặp ghép lớn nhất.

– Nếu $b \in h(B)$ thì $b \in h(B_2)$. Ký hiệu đỉnh $d' = h^{-1}(b) \in B_2$.

Đường đơn: $\langle (d', b) + (b, a) + (X) \rangle$ dẫn đỉnh d' trong B_2 tới đỉnh d nằm ngoài B . Vậy thì: $d' \in B_1$. Đó là điều mâu thuẫn.

2) Trường hợp: $a \notin B$. Khi đó $b \in h(B_2)$ vì (a, b) không tựa vào tập C .

Ký hiệu: $d' = h^{-1}(b) \in B_2$. Đường đơn $\langle (d', b) + (b, a) \rangle$ dẫn đỉnh d' tới đỉnh a ở ngoài tập B . Vậy thì $d \in B_1$. Suy ra điều mâu thuẫn.

Vậy C là một tập tựa của đồ thị. Tập tựa này có số phần tử bằng số phần tử của cặp ghép lớn nhất W .

Theo ký hiệu của Định lý 5.1: $k \leq$ số phần tử của $C =$ số phần tử của cặp ghép lớn nhất W . Định lý được chứng minh. ■

Với đồ thị hai phần $G = (V_1, V_2, F)$, ký hiệu:

$$d_0 = \max \{ |B| - |F(B)| \mid B \subseteq V_1 \}$$

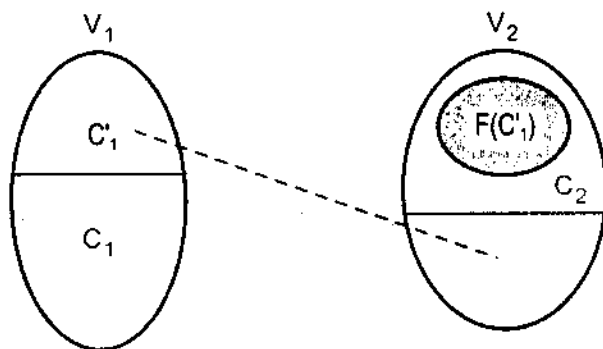
Vì $\emptyset \subset V_1$ và $|\emptyset| - |F(\emptyset)| = 0 - 0 = 0$ nên d_0 luôn là một số không âm.

Định lý 5.2: Số phần tử của tập tựa bé nhất của đồ thị hai phần $G = (V_1, V_2, F)$ là bằng $|V_1| - d_0$.

Chứng minh: Giả sử C là một tập tựa bất kỳ.

Có thể viết $C = C_1 \cup C_2$, trong đó $C_1 \subseteq V_1$ và $C_2 \subseteq V_2$.

Ký hiệu: $C_1' = V_1 \setminus C_1$. Khi đó, $F(C_1') \subseteq C_2$, vì nếu ngược lại thì sẽ có đỉnh $a \in C_1'$ mà đỉnh kề của nó $y \notin C_2$ và cạnh (a, y) sẽ không tựa vào tập C , dẫn tới mâu thuẫn.



Hình 5.8. Cách tìm tập tựa bé nhất

Mặt khác, $C_1 \cup F(C_1')$ lại là một tập tựa và $C_1 \cup F(C_1') \subseteq C$. Do đó, với mỗi tập tựa C có thể thay bằng một tập tựa dạng $C_1 \cup F(C_1')$ với số phần tử không lớn hơn.

Vậy để xét tập tựa bé nhất chỉ cần xét:

$$\begin{aligned} & \min \{ |C_1 \cup F(C_1')| \mid C_1 \subseteq V_1 \} = \\ & \min \{ |C_1| + |F(C_1')| \mid C_1 \subseteq V_1 \} = \\ & |V_1| - \max \{ |C_1'| - |F(C_1')| \mid C_1' \subseteq V_1 \} = |V_1| - d_0. \blacksquare \end{aligned}$$

Từ định lý này ta gọi d_0 là số hụt của đồ thị.

Hệ quả 5.1:

- Số ổn định trong của đồ thị hai phần G bằng $|V_2| + d_0$.
- Số phần tử của cặp ghép lớn nhất của G bằng $|V_1| - d_0$.

Ví dụ 5.4 (Bài toán phân công nhiệm vụ):

Một cơ quan có n nhân viên $x_1, x_2, \dots, x_n$ và m nhiệm vụ $y_1, y_2, \dots, y_m$. Do quá trình đào tạo, mỗi nhân viên có thể đảm nhiệm k nhiệm vụ và mỗi nhiệm vụ có đúng k nhân viên có thể đảm nhiệm được. Chứng minh rằng, luôn có thể phân công mỗi nhân viên đảm nhiệm một nhiệm vụ thích hợp với trình độ của người đó.

Thật vậy, ký hiệu V_1 là tập nhân viên và V_2 là tập các nhiệm vụ. Thế thì:

$$|V_1| = n \text{ và } |V_2| = m.$$

Xây dựng đồ thị hai phần $G = (V_1, V_2, F)$ mà:

$$x_i \in F(y_j) \Leftrightarrow \text{nhan vien } x_i \text{ đảm nhiệm được nhiệm vụ } y_j.$$

Theo giả thiết của bài toán thì mỗi đỉnh kề với đúng k cạnh.

Với $B \subseteq V_1$ thì số cạnh kề B sẽ là $k \cdot |B|$ và số cạnh kề $F(B)$ là $k \cdot |F(B)|$.

Số cạnh kề $F(B) \geq$ số cạnh kề B vì cạnh kề B thì cũng kề $F(B)$.

Cho nên, $|B| \leq |F(B)|$. Suy ra $d_0 = 0$. Theo Hệ quả 5.1, lực lượng của cặp ghép lớn nhất sẽ là $|V_1| - d_0 = |V_1|$. Do vậy, có thể phân công

cho n nhân viên đảm nhiệm n nhiệm vụ. Bằng cách thay đổi vai trò giữa V_1 và V_2 suy ra lực lượng của cặp ghép lớn nhất bằng $|V_2|$. Vậy $|V_1| = |V_2|$, nghĩa là số nhân viên bằng số nhiệm vụ. Bài toán phân công được giải quyết xong.

Đồ thị hai phần và cặp ghép của nó có nhiều ứng dụng trong thực tế. Vậy khi nào có thể tách từ một đồ thị vô hướng ra một đồ thị riêng hai phần với tập các cạnh của nó là một cặp ghép? Ta có kết quả sau:

Định lý 5.3: Đồ thị vô hướng $G = (V, E)$, với $|V| = 2n$ và bậc của mỗi đỉnh của đồ thị không ít hơn n , luôn có một đồ thị riêng hai phần $G'' = (V_1, V_2, E'')$ với $|V_1| = |V_2| = |E''| = n$ và tập các cạnh E'' chính là một cặp ghép lớn nhất của đồ thị G .

Chứng minh: Xây dựng đồ thị riêng hai phần $G'' = (V_1, V_2, E'')$ như sau:

Lấy dần vào tập E'' các cạnh của G sao cho các đỉnh trên các cạnh này khác nhau từng đôi cho đến khi bất kỳ cạnh nào còn lại cũng kề với cạnh nào đó thuộc E'' . Giả sử $|E''| = k$.

- Nếu $k = n$ thì định lý được chứng minh.
- Nếu $k < n$ thì $|V| \geq 2k + 2$.

Ký hiệu k cạnh thuộc E'' là: $(a_1, a_2), (a_3, a_4), \dots, (a_{2k-1}, a_{2k})$. Sẽ còn ít nhất hai đỉnh a_{2k+1}, a_{2k+2} không nằm trên cạnh nào thuộc E'' .

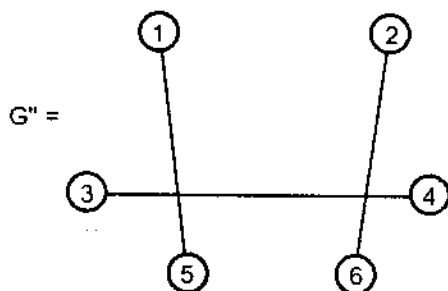
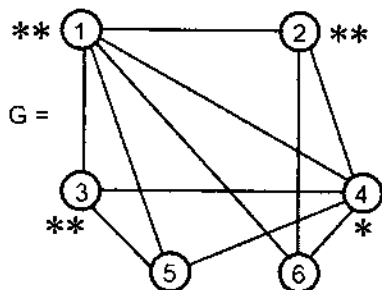
Theo cách chọn tập cạnh E'' thì hai đỉnh a_{2k+1}, a_{2k+2} chỉ kề với các đỉnh trên E'' , vì nếu chúng kề với các đỉnh không nằm trên E'' thì đã có thể bổ sung cho E'' một cạnh mới nữa. Vậy theo giả thiết, mỗi đỉnh a_{2k+1}, a_{2k+2} kề với ít nhất n đỉnh trên E'' .

Đánh dấu các đỉnh trên E'' kề với a_{2k+1} . Số đỉnh được đánh dấu $\geq n$. Ta lại đánh dấu các đỉnh trên E'' mà đỉnh cùng cạnh với nó trong E'' kề với a_{2k+2} . Số đỉnh được đánh dấu $\geq n$. Như vậy, số lần đánh dấu trên các đỉnh thuộc $E'' \geq 2n$. Nhưng số đỉnh trên E'' là $2k < 2n$ nên sẽ có ít nhất một đỉnh được đánh dấu hai lần.

Giả sử đỉnh này là a_i và đỉnh cùng cạnh với nó trong E'' là a_j . Đỉnh a_i kề với a_{2k+1} , còn a_j kề với a_{2k+2} . Như vậy, có thể loại cạnh (a_i, a_j) ra khỏi E'' và thêm vào hai cạnh mới (a_i, a_{2k+1}) và (a_j, a_{2k+2}) . Số cạnh trong E'' tăng thêm một. Cứ tiếp tục như thế, sau một số bước ta có $|E''| = n$ và xây dựng được đồ thị hai phần G'' . ■

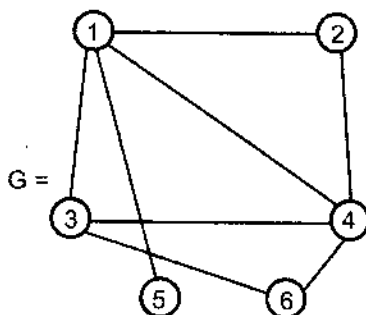
Ví dụ 5.5:

a)



Hình 5.9. Đồ thị và đồ thị riêng hai phần

b)



Không có

Hình 5.10. Đồ thị không có đồ thị riêng hai phần

BÀI TẬP

- 5.1. Chứng minh rằng, một đồ thị là đồ thị hai phần nếu và chỉ nếu mọi chu trình của nó đều có độ dài chẵn.
- 5.2. Cho một số đồ thị rồi tìm cặp ghép lớn nhất trong từng đồ thị.
- 5.3. Cho một số đồ thị rồi kiểm tra xem chúng có phải là đồ thị hai phần hay không.

Chương 6

CÁC THUẬT TOÁN DUYỆT ĐỒ THỊ

Phép duyệt đồ thị là một cách liệt kê tất cả các đỉnh của đồ thị này thành một danh sách tuyến tính. Hay nói một cách khác, phép duyệt đồ thị cho ta một cách "đi qua" tất cả các đỉnh của đồ thị để truy nhập, thêm bớt thông tin ở các đỉnh của đồ thị đó.

Phép duyệt đồ thị không phụ thuộc vào hướng của các cạnh. Do vậy, với đồ thị có hướng thì cần vô hướng hoá trước khi duyệt.

6.1. CÁC THUẬT TOÁN DUYỆT ĐỒ THỊ

6.1.1. Thuật toán duyệt đồ thị

Giả sử $G = (V, E)$ là đồ thị đã cho, x_0 là một đỉnh nào đó của G . Ký hiệu DS là một cấu trúc dữ liệu kiểu danh sách dùng để chứa các đỉnh.

- 1) Khởi đầu: $DS \leftarrow \{x_0\}$;
- 2) Lấy đỉnh x ra khỏi đầu DS;
- 3) Duyệt đỉnh x ;
- 4) Nạp các đỉnh của $F(x)$ vào DS;
- 5) Nếu $DS \neq \emptyset$ thì quay lên bước 2;
- 6) Dừng.

6.1.2. Duyệt đồ thị theo chiều sâu (Depth-First Search)

Nếu trong thuật toán trên, danh sách DS được tổ chức theo kiểu stack (danh sách vào sau – ra trước – LIFO) thì ta có phương pháp duyệt theo chiều sâu.

Trong phương pháp này, mỗi lần duyệt một đỉnh ta duyệt đến tận cùng mỗi nhánh rồi mới chuyển sang duyệt nhánh khác.

Giả sử $G = (V, E)$ là một đồ thị vô hướng.

Bắt đầu duyệt từ một đỉnh x_0 nào đó của đồ thị. Sau đó chọn x là đỉnh kề nào đó của x_0 và lặp lại quá trình duyệt đối với đỉnh x .

Giả sử ta đang xét đỉnh x :

– Nếu trong số các đỉnh kề với x , tìm được đỉnh y chưa được duyệt thì sẽ xét đỉnh này và bắt đầu từ đó tiếp tục quá trình duyệt. Ngược lại, nếu không còn đỉnh nào kề với x chưa được duyệt thì ta nói rằng đỉnh x đã duyệt xong và quay trở lại tiếp tục duyệt từ đỉnh mà từ đó ta đến được đỉnh x .

– Nếu quay trở lại đúng đỉnh x_0 thì phép duyệt kết thúc.

Thuật toán 6.1 (Duyệt đồ thị theo chiều sâu):

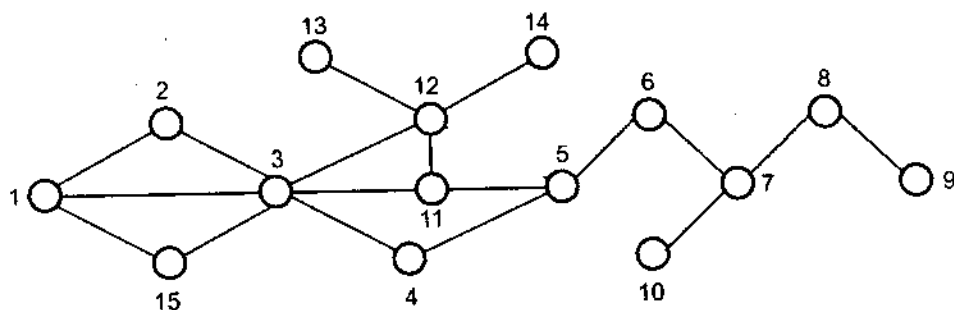
Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G .

Kết quả: Danh sách các đỉnh của đồ thị G .

```
1  procedure D_SAU (v) ;
2  begin
3    Thăm_đỉnh (v) ;
4    Duyệt [ v ] := true ;
5    for u ∈ DK[ v ] do
6      if ! Duyệt [ u ] then D_SAU (u) ;
7    end ;
8  BEGIN                                { Chương trình chính }
9    for v ∈ V do Duyệt [ v ] := false ;
10   for v ∈ V do
11     if ! Duyệt [ v ] then D_SAU (v) ;
12 END.
```

Độ phức tạp của thuật toán là: $O(n + m)$.

Ví dụ 6.1: Đồ thị được duyệt theo chiều sâu.



Hình 6.1. Thứ tự của các đỉnh được duyệt theo chiều sâu

Trong thuật toán duyệt theo chiều sâu, đỉnh được thăm càng muộn càng sớm trở thành duyệt xong. Do vậy, việc dùng một ngăn xếp (*stack*) để lưu trữ các đỉnh đang duyệt là rất thích hợp. Ta có thủ tục cải tiến sau đây:

Thuật toán 6.2 (Duyệt đồ thị theo chiều sâu):

```

1  procedure D_SAU_2 (v) ;
2  begin
3    S := Ø ;
4    Thăm_đỉnh (v) ;
5    Duyệt [ v ] := true ;
6    push v onto S ; { Nạp v lên đỉnh của S }
7    while S ≠ Ø do
8      begin
9        while '∃ u ∈ DK[top(S)] do
10         if ! Duyệt [ u ] then
11           begin
12             Thăm_đỉnh (u) ;
13             Duyệt [ u ] := true ;
14             push u onto S ;
15           end ;
16       pop(S) ; { Loại bỏ phần tử ở đỉnh của S }
17     end ;
18   end ;

```

6.1.3. Duyệt đồ thị theo chiều rộng (Breadth-First Search)

Nếu trong thuật toán duyệt đồ thị, cấu trúc danh sách DS được tổ chức theo kiểu hàng đợi (*danh sách vào trước – ra trước – FIFO*) thì ta có phương pháp duyệt theo chiều rộng. Trong phương pháp này, việc duyệt có tính chất "lan rộng". Một đỉnh được duyệt xong ngay sau khi ta đã xét hết tất cả các đỉnh kề với nó. Đỉnh được xét càng sớm thì sớm trở thành duyệt xong.

Thuật toán 6.3 (Duyệt đồ thị theo chiều rộng):

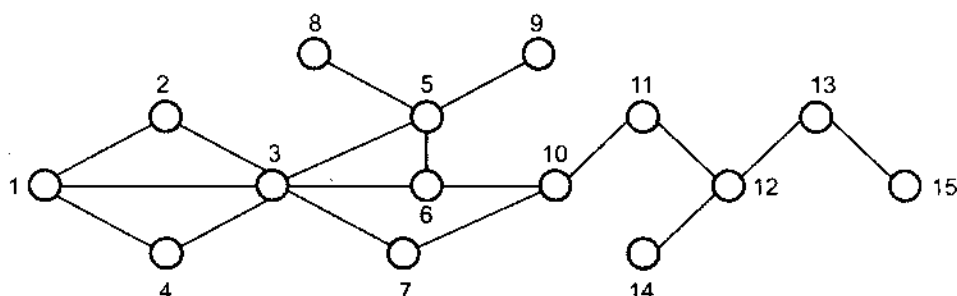
Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G .

Kết quả: Danh sách các đỉnh của đồ thị G .

```
1  procedure D_RONG (v) ;
2  begin
3      Q := Ø ;
4      enqueue v into Q ;
      { Nạp v vào cuối hàng đợi Q }
5      Duyệt [ v ] := true ;
6      while Q ≠ Ø do
7          begin
8              dequeue z from Q ;
              { Loại z ra khỏi đầu hàng đợi Q }
9              Thăm_đỉnh (z) ;
10             for u ∈ DK[ z ] do
11                 if ! Duyệt [ u ] then
12                     begin
13                         enqueue u into Q ;
14                         Duyệt [ u ] := true ;
15                     end ;
16             end ;
17         end ;
18     BEGIN { Chương trình chính }
19     for v ∈ V do Duyệt [ v ] := false ;
20     for v ∈ V do
21         if ! Duyệt [ v ] then D_RONG (v) ;
22     END.
```

Thuật toán này cũng có độ phức tạp là: $O(n + m)$.

Ví dụ 6.2: Đồ thị trong Ví dụ 6.1 được duyệt theo chiều rộng.



Hình 6.2. Thứ tự của các đỉnh được duyệt theo chiều rộng

6.2. MỘT SỐ ỨNG DỤNG CỦA THUẬT TOÁN DUYỆT ĐỒ THỊ

6.2.1. Bài toán tìm đường đi

Giả sử a và b là hai đỉnh nào đó của đồ thị vô hướng G . Hãy tìm đường đi (nếu có) từ đỉnh a đến đỉnh b .

Bài toán này đã được giải quyết bằng Thuật toán 1.1 hoặc thuật toán Warshall. Theo các phương pháp duyệt đồ thị, các lời gọi thủ tục $D_SAU(a)$ hoặc $D_RONG(a)$ cho phép thăm tất cả các đỉnh thuộc cùng thành phần liên thông với đỉnh a . Vì vậy, sau khi thực hiện xong thủ tục nếu $Duyet[b] = \text{false}$ thì không có đường đi từ đỉnh a đến đỉnh b , ngược lại thì đỉnh b thuộc cùng mảng liên thông với đỉnh a , hay nói một cách khác, có đường đi từ a đến b .

Để khôi phục đường đi, dùng thêm một biến mảng $Truoc$. Thành phần $Truoc[u]$ ghi lại đỉnh đến trước đỉnh u trên đường duyệt từ a tới u . Khi đó, trong thủ tục $D_SAU(v)$ cần sửa đổi câu lệnh if ở dòng lệnh 6 như sau:

```

6   if !Duyet [ u ] then
      begin
          Truoc [ u ] := v ;
          D_SAU ( u )
      end ;

```

còn trong thủ tục D_RONG (v) thì cần sửa câu lệnh `if` ở các dòng lệnh 11–15 là:

```
11  if  !Duyet [  $u$  ]  then
12      begin
13          enqueue   $u$  into   $Q$  ;
14          Duyet [  $u$  ] := true ;
           Truoc [  $u$  ] :=  $z$ 
15      end ;
```

Đường đi cần tìm (nếu có) sẽ được khôi phục như sau:

$$b \leftarrow a_I = \text{Truoc}[b] \leftarrow a_2 = \text{Truoc}[a_I] \leftarrow \dots \leftarrow a$$

Chú ý: Đường đi tìm được theo thuật toán duyệt theo chiều rộng là đường đi ngắn nhất từ đỉnh a đến đỉnh b .

6.2.2. Bài toán tìm các mảng liên thông

Cho đồ thị G , hãy tìm số mảng liên thông p của đồ thị này và xác định xem mỗi mảng liên thông bao gồm những đỉnh nào.

Do các thủ tục D_SAU(v) hoặc D_RONG(v) cho phép duyệt tất cả các đỉnh thuộc cùng mảng liên thông với đỉnh v , nên số mảng liên thông p của đồ thị G chính bằng số lần gọi đến các thủ tục này trong chương trình chính.

Để ghi nhận các đỉnh trong từng mảng liên thông, dùng thêm biến mảng $\text{Mang}[v]$ để ghi chỉ số của mảng liên thông chứa đỉnh v . Chỉ số này tăng từ 1 đến p .

Dùng biến p để đếm số mảng liên thông của đồ thị và gán chỉ số cho các mảng liên thông tìm được. Bắt đầu nó được khởi tạo giá trị bằng 0. Thủ tục $\text{Thăm_đỉnh}(v)$ trong các thủ tục D_SAU(v) và D_RONG(v) làm thêm nhiệm vụ gán:

$$\text{Mang}[v] := p ;$$

Còn chương trình chính của thuật toán duyệt cần được sửa lại như sau:


```

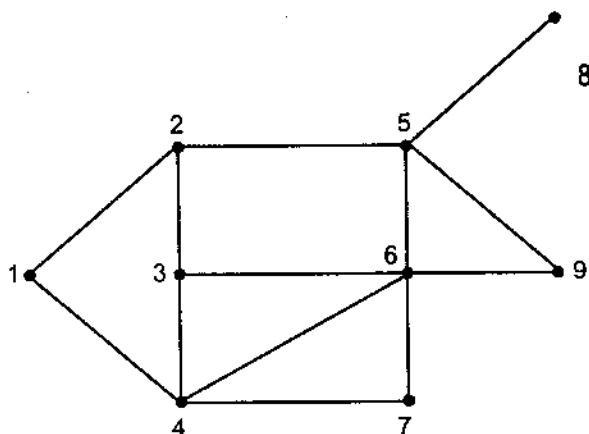
1 BEGIN  { Chương trình chính }
2   for v ∈ V do
3       Duyet [ v ] := false ;
4   p := 0 ;
5   for v ∈ V do
6       if ! Duyet [ v ] then
7           begin
8               p := p + 1 ;
9               D_SAU (v) ; { D_RONG (v) ; }
10          end ;
11   END.

```

Khi chương trình kết thúc, biến p sẽ cho số mảng liên thông của đồ thị, còn các giá trị của biến mảng $Mang[v]$, $v \in V$ cho phép liệt kê tất cả các đỉnh trong từng mảng liên thông của đồ thị.

BÀI TẬP

- 6.1. Cho ba đồ thị, sau đó duyệt theo chiều sâu, rồi duyệt theo chiều rộng trên các đồ thị này.
- 6.2. Từ kết quả của phép duyệt theo chiều sâu và duyệt theo chiều rộng của một đồ thị, hãy xây dựng lại đồ thị này.
- 6.3. Duyệt theo chiều sâu, rồi duyệt theo chiều rộng đồ thị sau:



Hình 6.3. Đồ thị vô hướng 9 đỉnh

Chương 7

CHU TRÌNH EULER VÀ CHU TRÌNH HAMILTON

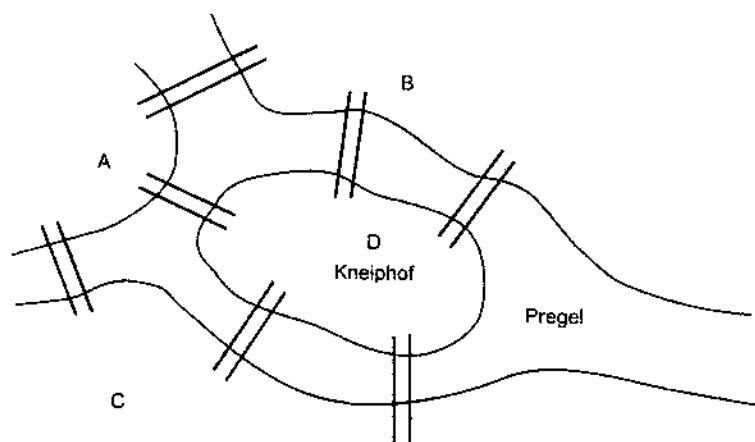
Chu trình Euler và chu trình Hamilton là hai loại chu trình rất nổi tiếng trong Lý thuyết Đồ thị, mà tên gọi của chúng gắn với tên của các nhà khoa học tìm ra nó. Không những thế, chúng còn nổi tiếng vì một số bài toán liên quan vẫn còn là những bài toán mở.

7.1. CHU TRÌNH EULER

Khái niệm chu trình Euler được ra đời từ bài toán nổi tiếng sau đây:

Ví dụ 7.1 (Bài toán 7 cây cầu ở Königsberg):

Thành phố Königsberg thuộc nước Cộng hoà Litva có con sông Pregel chảy qua, giữa sông có cù lao Kneiphof tạo nên bốn vùng đất. Người ta đã xây dựng 7 cây cầu để nối các vùng đất này lại với nhau như Hình 7.1



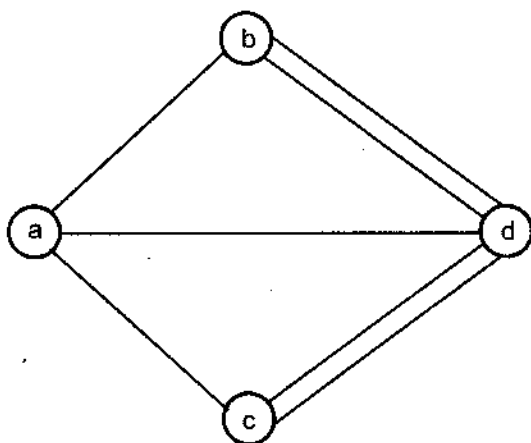
Hình 7.1. Bảy cây cầu trên sông Pregel

Năm 1736, L. Euler nhà toán học người Thụy Sĩ đã đưa ra bài toán sau đây:

Hỏi có thể đi qua cả 7 cây cầu, mỗi cầu chỉ đi qua một lần rồi quay trở về đúng chỗ xuất phát được hay không?

Bài toán đã làm say mê cư dân của thành phố. Họ háo hức đi thử nhưng không thành công. Sau đó, chính L. Euler đã chứng minh được rằng, bài toán trên là không giải được.

Nếu biểu diễn mỗi vùng đất: A, B, C, D bằng đỉnh của một đa đồ thị vô hướng và có cạnh nối giữa chúng nếu có cầu nối tương ứng, thì bài toán trên đưa về việc tìm một chu trình đi qua mỗi cạnh của đồ thị đúng một lần.



Hình 7.2. Đa đồ thị biểu diễn thành phố Königsberg

Từ bài toán trên, người ta đã đưa ra các khái niệm sau:

Định nghĩa 7.1: Đường Euler của đa đồ thị là đường đi qua mỗi cạnh của đồ thị đúng một lần. Chu trình Euler của đa đồ thị là chu trình đi qua mỗi cạnh của đồ thị đúng một lần.

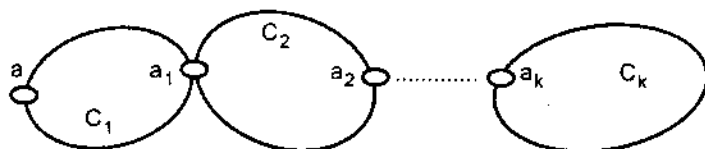
Từ định nghĩa trên, ta có điều kiện cần và đủ cho sự tồn tại của chu trình vô hướng Euler như sau:

Định lý 7.1: Đa đồ thị liên thông G có chu trình vô hướng Euler khi và chỉ khi mỗi đỉnh đều có bậc chẵn.

Chứng minh:

$\Rightarrow$: Mỗi lần chu trình đi qua một đỉnh thì đỉnh đó bớt đi hai cạnh kề. Cuối cùng số cạnh kề của mỗi đỉnh bằng 0. Do đó số cạnh kề với mỗi đỉnh phải là một số chẵn.

$\Leftarrow$: Xuất phát từ một đỉnh a nào đó, lập dãy cạnh kề liên tiếp cho đến khi hết khả năng đi tiếp. Khi dừng phải ở đỉnh a vì bậc các đỉnh đều chẵn, ta được chu trình C_1 . Nếu đã vét hết các cạnh thì đó chính là chu trình cần tìm. Nếu vẫn còn cạnh, do tính liên thông của đồ thị thì phải có một cạnh nào đó chưa được chọn và kề với đỉnh a_1 nào đó của chu trình đã có. Lại xuất phát từ a_1 , và tiếp tục quá trình như trên cho đến khi hết khả năng đi tiếp, ta được chu trình $C_2, \dots$



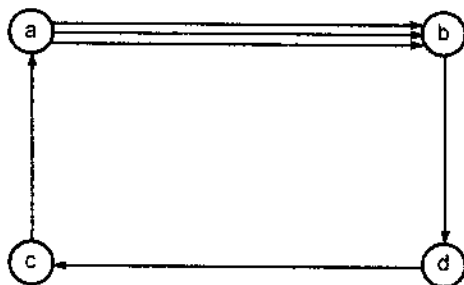
Hình 7.3. Các chu trình kề nhau

Khi đã vét hết các cạnh, lập chu trình Euler cho đồ thị này như sau:

Từ đỉnh a đi theo nửa trên của C_1 cho đến a_1 , lại tiếp tục từ a_1 đi theo nửa trên của C_2 cho đến $a_2, \dots$ Khi đã đến chu trình con cuối cùng, đi ngược lại theo các nửa dưới của các chu trình con... và cuối cùng trở về đỉnh a . Ta nhận được một chu trình Euler. ■

Đa đồ thị có hướng có thể có chu trình Euler vô hướng nhưng không có chu trình Euler có hướng.

Ví dụ 7.2:



Hình 7.4. Đa đồ thị có hướng

Hệ quả 7.1: Đa đồ thị liên thông G có đường đi Euler vô hướng khi và chỉ khi số đỉnh bậc lẻ bằng 2.

Chứng minh:

$\Rightarrow$: Nếu có đường đi Euler vô hướng nối a với b thì a, b là hai đỉnh duy nhất có bậc lẻ.

$\Leftarrow$: Giả sử a, b là hai đỉnh duy nhất có bậc lẻ. Xây dựng đồ thị G' từ G bằng cách thêm vào cạnh (a, b) . G' sẽ không có đỉnh bậc lẻ, nên theo Định lý 7.1 G' sẽ có chu trình Euler vô hướng. Bỏ cạnh (a, b) , ta có đường đi Euler vô hướng trong đồ thị G . ■

Với đồ thị có hướng, ta có điều kiện cần và đủ cho sự tồn tại của chu trình có hướng Euler như sau.

Định lý 7.2: Đa đồ thị có hướng liên thông có chu trình Euler có hướng khi và chỉ khi tại mỗi đỉnh của đồ thị số cạnh đi vào bằng số cạnh đi ra (đỉnh cân bằng), nghĩa là:

$$\forall x \in V, r^-(x) = r^+(x),$$

trong đó, $r^-(x)$ là số cạnh đi vào đỉnh x và $r^+(x)$ là số cạnh đi ra khỏi đỉnh x .

Hệ quả 7.2: Đa đồ thị có hướng liên thông G có đường đi Euler có hướng khi và chỉ khi trong G có hai đỉnh a, b thoả mãn:

$$r^-(a) = r^+(a) - 1,$$

$$r^-(b) = r^+(b) + 1$$

và các đỉnh còn lại đều cân bằng.

Chứng minh:

1) Giả sử đồ thị G có đường Euler α đi qua tất cả các cạnh của đồ thị. Khi đó thì:

– Với đỉnh xuất phát a của α : trừ cạnh đầu tiên của α đi ra từ a , cứ có một cạnh đi vào a thì phải có một cạnh đi ra khỏi a vì α kết thúc ở đỉnh khác. Do vậy:

$$r^-(a) = r^+(a) - 1.$$

– Với đỉnh kết thúc b của α : trừ cạnh cuối cùng của α đi tới b , cứ có một cạnh đi ra khỏi b thì phải có một cạnh đi vào b vì α kết thúc ở đỉnh b . Vậy thì:

$$r^-(b) = r^+(b) + 1.$$

– Với các đỉnh khác: số cạnh đi vào bằng số cạnh đi ra nên chúng là cân bằng.

2) Giả sử đồ thị G liên thông và có hai đỉnh a, b có tính chất trên. Thêm vào một cạnh mới (b, a) . Khi đó mọi đỉnh là cân bằng nên theo Định lý 7.2 đồ thị có chu trình Euler có hướng. Bây giờ, bỏ cạnh (b, a) khỏi chu trình, ta được một đường đi Euler có hướng. ■

Dựa trên chứng minh của Định lý 7.1, xây dựng thuật toán tìm chu trình Euler cho đồ thị liên thông không có đỉnh bậc lẻ như sau:

Thuật toán 7. 1 (Tìm chu trình Euler):

Dữ liệu: Đồ thị liên thông $G = (V, E)$ không có đỉnh bậc lẻ được biểu diễn bởi mảng các danh sách kề DK.

Kết quả: Chu trình vô hướng Euler với danh sách các đỉnh nằm trong stack CE.

```

1  Begin
2  S := Ø ; CE := Ø ;
3  v := đỉnh tùy ý của đồ thị ;
4  push v onto S ;
5  while S ≠ Ø do
6    begin
7      v := top(S) ;
8      if DK(v) ≠ Ø then
9        begin
10         u:=đỉnh đầu tiên trong danh sách DK[ v] ;
11         push u onto S ;
12         DK[ v] := DK[ v] \ { u} ;

```

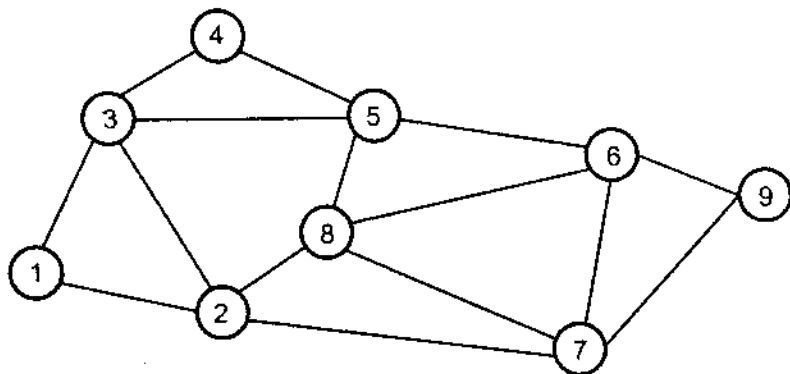
```

12      DK[ v ] := DK[ v ] \ { u } ;
13      DK[ u ] := DK[ u ] \ { v } ;
          { Xoá cạnh (v, u) trong đồ thị G }
14      v := u
15  end
16  else
17      begin
18          v := top(S) ;
19          push v onto CE
20      end
21  end
22  End.

```

Ta thấy rằng, mỗi lần lặp của chu trình (từ câu lệnh 5 đến câu lệnh 20) trong thuật toán, hoặc là đặt đỉnh lên stack S và xoá cạnh, hoặc chuyển đỉnh từ stack S sang stack CE. Số lần lặp của chu trình không vượt quá số cạnh m . Vậy độ phức tạp tổng thể của thuật toán là $O(m)$. Đây là một thuật toán tối ưu để tìm chu trình Euler.

Ví dụ 7.3: Áp dụng thuật toán trên cho đồ thị vô hướng với các đỉnh bậc chẵn dưới đây:



Hình 7.5. Đồ thị vô hướng với các đỉnh bậc chẵn

Ta nhận được chu trình Euler là:

[1, 2, 3, 4, 5, 6, 7, 2, 8, 6, 9, 7, 8, 5, 3, 1].

7.2. CHU TRÌNH HAMILTON

Năm 1857, W. R. Hamilton nhà toán học người Ailen đã đưa ra trò chơi sau đây: Trên mỗi đỉnh trong số 20 đỉnh của một khối đa diện 12 mặt ngũ giác đều có ghi tên một thành phố lớn của thế giới. Hãy tìm cách đi bằng các cạnh của khối này để đi qua tất cả các thành phố, mỗi thành phố đúng một lần.

Bài toán này đã dẫn tới những khái niệm sau:

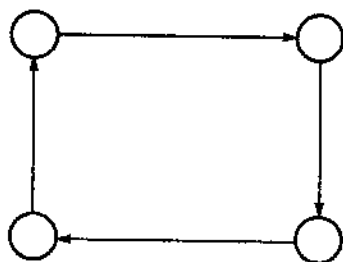
Định nghĩa 7.2: Đường Hamilton là đường đi qua mỗi đỉnh của đồ thị đúng một lần. Chu trình Hamilton là chu trình đi qua mỗi đỉnh của đồ thị đúng một lần.

Ví dụ 7.4:

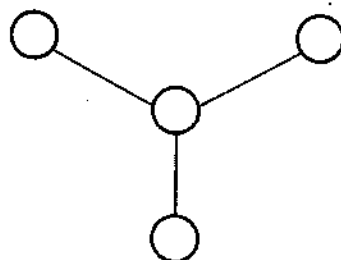
1. Tổ chức tour du lịch sao cho người du lịch thăm quan mỗi thắng cảnh trong thành phố đúng một lần.
2. Cho quân mã đi trên bàn cờ vua sao cho nó đi qua mỗi ô đúng một lần (*bài toán mã đi tuần*).

Với một đồ thị đã cho, đường (chu trình) Hamilton có thể tồn tại hoặc không.

Ví dụ 7.5:



a) Đồ thị có chu trình Hamilton



b) Đồ thị không có đường Hamilton

Hình 7.6. Đồ thị có và đồ thị không có chu trình Hamilton

Định lý 7.3 (Rédei): Đồ thị đầy đủ luôn có đường đi Hamilton.

Chứng minh:

Xét đồ thị có hướng G .

Ta chứng minh bằng quy nạp theo số đỉnh n của G .

$n = 1, 2$: Hiển nhiên.

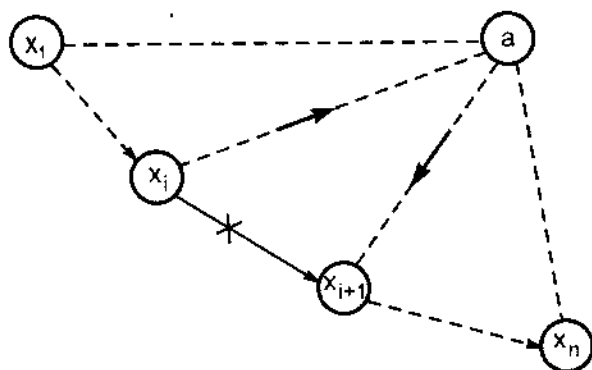
$(n) \Rightarrow (n + 1)$: Giả sử G là đồ thị đầy đủ có $n + 1$ đỉnh và đồ thị G' xây dựng từ G bằng cách bớt một đỉnh a và các cạnh kề với a . Đồ thị G' có n đỉnh và cũng đầy đủ nên theo giả thiết quy nạp, nó có đường Hamilton:

$$(H) = \langle x_1, x_2, \dots, x_n \rangle$$

Nếu trong G có cạnh (x_n, a) thì đường $\langle (H), a \rangle$ sẽ là đường Hamilton.

Nếu trong G có cạnh (a, x_1) thì đường $\langle a, (H) \rangle$ sẽ là đường Hamilton.

Trong trường hợp ngược lại, đồ thị G phải có các cạnh (a, x_n) và (x_1, a) , nghĩa là có các cạnh ngược hướng nhau. Khi đó sẽ phải có ít nhất một cặp cạnh sát nhau, nhưng ngược hướng nhau là (x_i, a) và (a, x_{i+1}) .



Hình 7.7. Cách tìm đường đi Hamilton

Ta có đường $\langle x_1, \dots, x_i, a, x_{i+1}, \dots, x_n \rangle$ là một đường đi Hamilton của G . ■

Đồ thị G được gọi là đồ thị có bậc 1 nếu tại mỗi đỉnh có đúng một cạnh vào và một cạnh ra. Hiển nhiên, chu trình Hamilton là một đồ thị riêng bậc 1 của đồ thị đã cho.

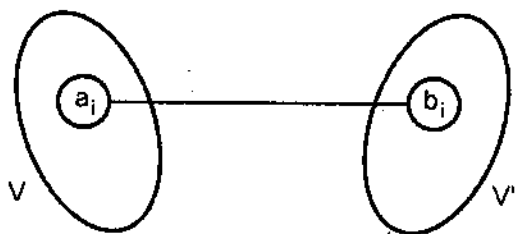
Định lý 7.4: Đồ thị $G = (V, F)$ có đồ thị riêng bậc 1 khi và chỉ khi:

$$\forall B \subseteq V, |B| \leq |F(B)|.$$

Chứng minh: Ký hiệu tập đỉnh của đồ thị là $V = \{a_1, a_2, \dots, a_n\}$. Lập tập $V' = \{b_1, b_2, \dots, b_n\}$ sao cho: $V \cap V' = \emptyset$. Xây dựng đồ thị hai phần $H = (V, V', F')$ với:

$$b_j \in F'(a_i) \Leftrightarrow a_j \in F(a_i)$$

Nếu đồ thị G có đồ thị riêng bậc 1 là $G^1 = (V, F^1)$ thì xét tập cạnh W của H như sau: $(a_i, b_j) \in W \Leftrightarrow a_j \in F^1(a_i)$. Đó là một cặp ghép n cạnh trong H .



Hình 7.8. Cách xây dựng đồ thị hai phần

Ngược lại, ứng với một cặp ghép n cạnh W trong H sẽ có một đồ thị riêng bậc 1 của G .

Theo Hệ quả 5.1 thì:

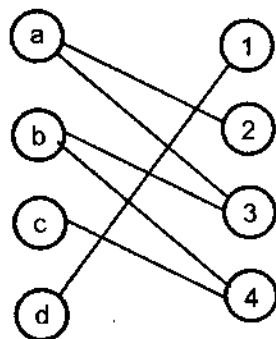
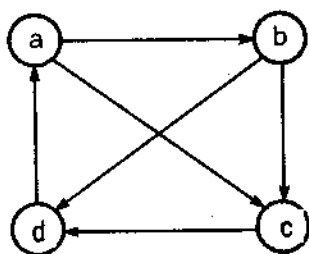
$$|W| = n \leq |V| - d_0$$

$$\Rightarrow d_0 = 0$$

$$\begin{aligned} \text{mà} \quad d_0 &= \max \{ |B| - |F(B)| \mid B \subseteq V \} = \\ &= \max \{ |B| - |F(B)| \mid B \subseteq V \} = 0 \end{aligned}$$

Suy ra: $|B| \leq |F(B)|$. Đó là điều phải chứng minh. ■

Ví dụ 7.6: Xét đồ thị có hướng $G = (V, F)$ được cho trong hình vẽ dưới đây:



Hình 7.9. Đồ thị định hướng và đồ thị hai phần tương ứng

Đặt $V' = \{1, 2, 3, 4\}$ và xây dựng đồ thị hai phần H . Chọn cặp ghép lớn nhất $W = \{(a, 2), (b, 3), (c, 4), (d, 1)\}$. Từ đó xây dựng được chu trình Hamilton cho đồ thị G là $[a, b, c, d]$.

Hệ quả 7.3: Nếu đồ thị G có chu trình Hamilton thì:

$$\forall B \subseteq V, |B| \leq |F(B)|.$$

Chứng minh: Vì chu trình Hamilton là đồ thị riêng bậc 1. Điều ngược lại là không đúng vì đồ thị riêng có thể gồm nhiều mảng liên thông rời nhau. ■

Từ Hệ quả này, có thể khẳng định rằng: Nếu trong đồ thị có một tập con B các đỉnh mà $|B| > |F(B)|$ thì đồ thị này không có chu trình Hamilton.

Hệ quả 7.4: Giả sử đồ thị có hướng G có đường đi Hamilton.

Ký hiệu: $d = \max \{|B| - |F(B)| \mid B \subseteq V\}$. Khi đó, số $d \in \{0, 1\}$.

Chứng minh: Giả sử đồ thị có hướng G có đường đi Hamilton $\langle a, \dots, b \rangle$.

Nếu trong đồ thị G có cạnh (b, a) thì G có chu trình Hamilton và theo Hệ quả 7.3 thì $d = 0$.

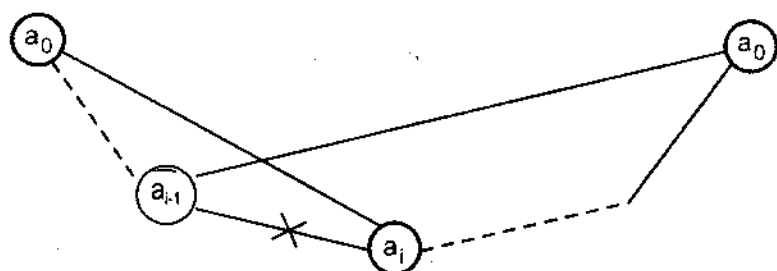
Ngược lại, giả sử đồ thị G không có cạnh (b, a) . Xét đồ thị G' xây dựng từ đồ thị G và thêm cạnh (b, a) . Thế thì đồ thị G' sẽ có $d' = 0$. Mặt khác, vì ta chỉ thêm một cạnh mà không thêm đỉnh, nên: $d \leq d' + 1$. Từ đó suy ra $d \leq 1$. ■

Cũng theo Hệ quả 7.3, có thể khẳng định rằng: Nếu đồ thị nào đó có $d \geq 2$ thì đồ thị ấy không có đường đi Hamilton.

Bổ đề 7.1: Giả sử đồ thị G có đường đi đơn vô hướng cực đại $\langle a_0, a_1, \dots, a_q \rangle$. Nếu $r(a_0) + r(a_q) \geq q + 1$ thì đồ thị con G' tạo bởi tập đỉnh $\{a_0, a_1, \dots, a_q\}$ có chu trình vô hướng Hamilton.

Chứng minh: Ký hiệu $r'(a)$ là bậc của đỉnh a trong G' . Vì đường đơn đã cho là cực đại nên:

$$r'(a_0) = r(a_0), r'(a_q) = r(a_q).$$



Hình 7.10. Cách tìm chu trình Hamilton

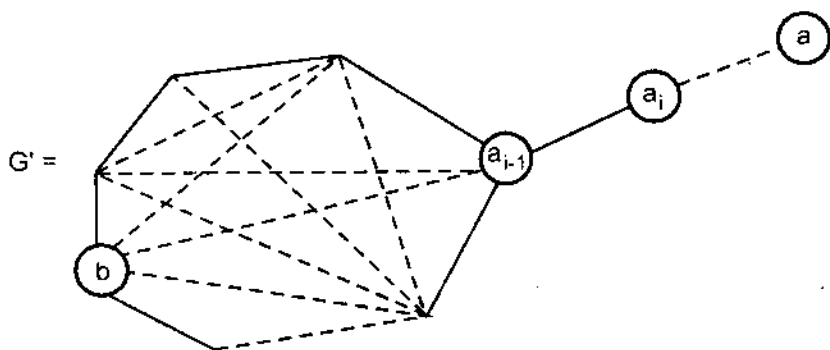
Giả sử $r(a_0) = k$, nghĩa là đỉnh a_0 kề với k đỉnh trên đường đi là $a_1, a_{i_2}, \dots, a_{i_k}$.

- Nếu a_0 kề với a_q thì G' có chu trình vô hướng Hamilton.
- Nếu như a_q không kề với $a_0, a_{i_2-1}, \dots, a_{i_k-1}$. Đó là những đỉnh nằm bên trái của dãy đỉnh trong đường đi nói trên. Thế thì: $r(a_q) \leq q - k$. Do đó: $r(a_0) + r(a_q) \leq q$, trái với giả thiết. Vậy phải có ít nhất một đỉnh a_{i-1} sao cho a_0 kề với a_i và a_q kề với a_{i-1} . Khi đó dãy các đỉnh $[a_0, a_i, \dots, a_q, a_{i-1}, \dots, a_0]$ là một chu trình vô hướng của G' . ■

Bổ đề 7.2: Giả sử đồ thị G liên thông. Nếu các đỉnh của đường đi đơn vô hướng dài nhất tạo nên một đồ thị con G' có chu trình vô hướng Hamilton thì chu trình đó cũng chính là chu trình vô hướng Hamilton của đồ thị G .

Chứng minh: Chỉ cần chứng minh rằng, đồ thị con G' chính là đồ thị G . Giả sử ngược lại, có đỉnh a thuộc G nhưng không thuộc đồ thị

con G' . Vì đồ thị G liên thông nên với đỉnh b thuộc G' sẽ có đường đi vô hướng trong G nối đỉnh b với đỉnh a là $\langle b = a_0, a_1, \dots, a \rangle$.



Hình 7.11. Đường đi nối đỉnh a với chu trình

Đỉnh b thuộc G' , còn đỉnh a không thuộc G' . Do vậy, sẽ có đỉnh a_i là đỉnh đầu tiên của đường đi không thuộc G' .

Xét đường đi sau đây trong đồ thị G : Lấy chu trình vô hướng Hamilton của G' , bỏ đi một cạnh kề với a_{i-1} và thêm vào cạnh (a_{i-1}, a_i) . Đường đi này có độ dài bằng số đỉnh của G' , trong khi đó đường đi đơn dài nhất đã cho trong G chỉ có độ dài là số đỉnh của G' bớt đi 1. Vậy trái với giả thiết. ■

Định lý 7.5: Giả sử đồ thị G có n đỉnh.

1) Nếu $\forall a, b \in V, r(a) + r(b) \geq n - 1$ thì G có đường đi vô hướng Hamilton.

2) Nếu $\forall a, b \in V, r(a) + r(b) \geq n$ thì G có chu trình vô hướng Hamilton.

Chứng minh:

1) Giả sử $\langle x_0, \dots, x_q \rangle$ là đường đi đơn vô hướng dài nhất của G . Đường đi này có độ dài là q .

– Nếu $q = n - 1$ thì đó là đường đi Hamilton của đồ thị G .

– Nếu $q \leq n - 2$ thì $r(x_0) + r(x_q) \geq n - 1 \geq q + 1$ và theo Bổ đề 7.1 đồ thị con G' tạo bởi tập đỉnh $\{x_0, \dots, x_q\}$ có chu trình vô hướng Hamilton. Giả sử chu trình vô hướng Hamilton đó là $[y_0, y_1, \dots, y_q]$.

Vì $q \leq n - 2$ nên có ít nhất một đỉnh y của G nằm ngoài chu trình trên.

Từ 1) suy ra đồ thị G liên thông. Vậy đỉnh y phải nối được với một đỉnh y_j nào đó trong chu trình bằng một đường đi đơn vô hướng.

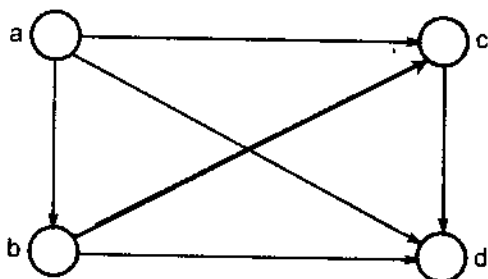
Ta nhận được đường đi $\langle y, \dots, y_j, y_{j-1}, \dots, y_0, y_q, \dots, y_{j+2}, y_{j+1} \rangle$ có độ dài lớn hơn $q + 1$. Vậy mâu thuẫn với tính cực đại của đường đi $\langle x_0, \dots, x_q \rangle$.

Do vậy $q = n - 1$. Đồ thị G luôn có đường đi vô hướng Hamilton.

2) Theo 1) thì trong đồ thị G có đường đi đơn vô hướng dài nhất chứa tất cả n đỉnh là $\langle y_0, y_1, \dots, y_{n-1} \rangle$.

Hơn nữa, $r(y_0) + r(y_{n-1}) \geq n = (n - 1) + 1$, nên theo Bổ đề 7.1 thì đồ thị con G' sinh bởi tập đỉnh $\{y_0, y_1, \dots, y_{n-1}\}$ có chu trình vô hướng Hamilton, và đó cũng chính là chu trình Hamilton của đồ thị G . ■

Ví dụ 7.7: Xét đồ thị có hướng sau đây:



Hình 7.12. Đồ thị có hướng có chu trình vô hướng Hamilton

Đồ thị thỏa mãn điều kiện 2) nên nó có chu trình vô hướng Hamilton.

Nếu bỏ cạnh (c, d) thì điều kiện 1) thỏa mãn, còn điều kiện 2) không thỏa mãn nữa. Đồ thị chỉ có đường đi vô hướng Hamilton.

Hệ quả 7.5 (Dirac): Nếu $\forall a \in V, r(a) \geq \frac{n}{2}$ thì đồ thị G có chu trình vô hướng Hamilton.

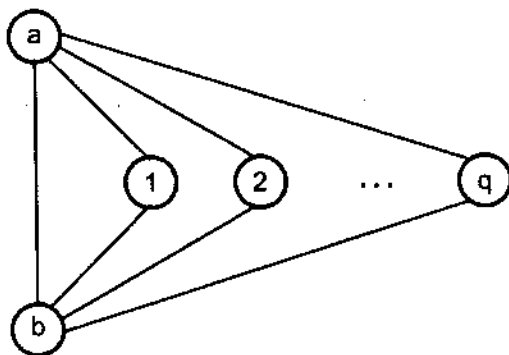
Chứng minh: Suy ra từ phần 2) của Định lý 7.5. ■

Một số nhận xét:

1. Đồ thị có đỉnh có bậc ≤ 1 thì không có chu trình Hamilton.
2. Đồ thị có các đỉnh đều có bậc ≥ 2 , nếu có đỉnh nào có bậc bằng 2 thì mọi chu trình Hamilton (nếu có) phải đi qua hai cạnh kề với đỉnh này.
3. Nếu trong đồ thị có đỉnh có ba đỉnh bậc 2 kề với nó thì đồ thị không có chu trình Hamilton.
4. Nếu đỉnh a có hai đỉnh kề bậc 2 là b và c thì mọi cạnh (a, x) , $x \notin \{b, c\}$ sẽ không thuộc bất kỳ chu trình Hamilton nào.
5. Đồ thị có đường đi vô hướng $\langle a_1, a_2, \dots, a_k \rangle$. Với chỉ số $k < n$ và các đỉnh trên đường đi (trừ a_1 và a_k) đều có bậc 2 thì cạnh (a_1, a_k) sẽ không thuộc bất kỳ chu trình Hamilton nào.
6. Đồ thị hai phần $G = (V_1, V_2, F)$ với $|V_1| \neq |V_2|$ sẽ không có một chu trình Hamilton nào.

BÀI TẬP

- 7.1. Tìm đồ thị có đường đi Euler nhưng không có đường đi Hamilton và ngược lại.
- 7.2. Tìm đồ thị có chu trình Euler nhưng không có chu trình Hamilton và ngược lại.
- 7.3. Tìm đồ thị có cả chu trình Euler và chu trình Hamilton.
- 7.4. Tìm đồ thị không có chu trình Euler lẫn chu trình Hamilton.
- 7.5. Cho đồ thị sau:

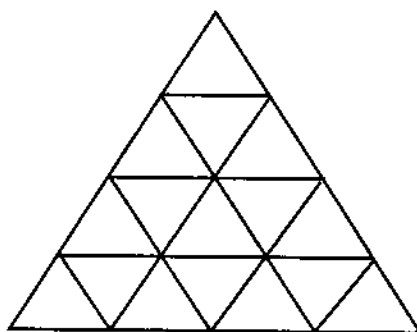


Hình 7.13. Đồ thị dãy tam giác

Với q là số nguyên lẻ, hỏi đồ thị này có bao nhiêu chu trình Euler?

- 7.6. Khu triển lãm Khoa học Công nghệ có các phòng triển lãm hình tam giác đều như Hình 7.14. Hai phòng triển lãm chung cạnh thì có cửa thông nhau.

Một khách tham quan muốn đi xem tất cả các phòng triển lãm. Liệu anh ta có thể đi qua tất cả các phòng, mỗi phòng đúng một lần được hay không?



Hình 7.14. Khu triển lãm

- 7.7. Đồ thị đường (line graph) $L(G)$ của đồ thị G là đồ thị có tập đỉnh là tập các cạnh của G và có cạnh nối hai đỉnh e_1, e_2 trong $L(G)$ nếu và chỉ nếu cạnh $e_1 = (x, y)$ và cạnh $e_2 = (y, z)$ trong đồ thị G . Chứng minh rằng:
- G có chu trình Euler khi và chỉ khi $L(G)$ có chu trình Hamilton.
 - Nếu G có chu trình Euler thì $L(G)$ cũng có chu trình Euler. Tìm một phản ví dụ chứng tỏ điều ngược lại không đúng.

Chương 8

BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT

Trước mỗi chuyến xuất hành, chúng ta thường phải suy nghĩ và chọn ra cho mình một hành trình "tiết kiệm" nhất theo nghĩa tốn ít thời gian, tốn ít nhiên liệu hoặc tốn ít tiền nhất... Lý thuyết Đồ thị sẽ giúp chúng ta tìm ra giải pháp đó.

8.1. BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT

Bài toán: Cho đồ thị $G = (V, E)$ và hai đỉnh a, b . Tìm đường đi ngắn nhất (nếu có) đi từ đỉnh a đến đỉnh b trong đồ thị G .

Ý nghĩa thực tế: Bài toán này giúp chúng ta chọn các hành trình tiết kiệm nhất (quãng đường, thời gian, chi phí...) trong giao thông, lập lịch thi công các công trình một cách tối ưu, xử lý trong truyền tin...

Thuật toán duyệt đồ thị theo chiều rộng đã cho lời giải của bài toán này. Song có thêm thuật toán sau:

Thuật toán 8.1:

1. Lần lượt gán nhãn cho các đỉnh của đồ thị, mỗi đỉnh không quá một lần, như sau:

- Đỉnh a được gán nhãn là số 0.
- Những đỉnh kề với đỉnh a được gán số 1.
- Những đỉnh kề với đỉnh đã được gán nhãn số 1, được gán số 2.

.....

– Tương tự, những đỉnh kề với đỉnh đã được gán số i được gán nhãn là số $i + 1$.

.....

Thực hiện cho đến khi gán được nhãn cho đỉnh b hoặc không gán nhãn được nữa.

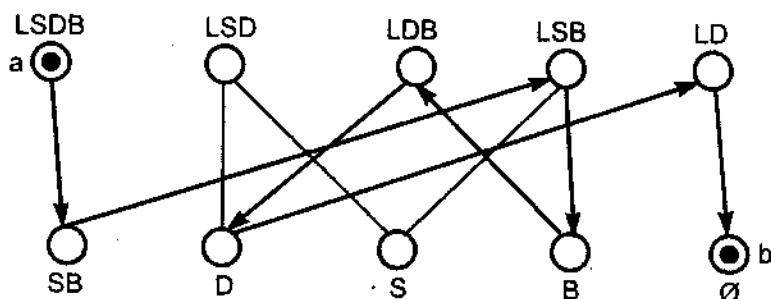
2. Nếu đỉnh b được gán nhãn nào đó là k thì kết luận có đường đi ngắn nhất từ đỉnh a tới đỉnh b với độ dài k , ngược lại thì trả lời là không có.

3. *Khôi phục đường đi*: Nếu ở bước 2 chỉ ra b được gán nhãn k nào đó thì ta đi ngược lại theo quy tắc sau: Nếu đỉnh y được gán nhãn j với $j \geq 1$ thì sẽ có đỉnh x được gán nhãn $j - 1$ sao cho có cạnh đi từ x tới y . Đi ngược lại cho đến khi gặp đỉnh a , ta nhận được đường đi ngắn nhất cần tìm.

Ví dụ 8.1 (*Bài toán con sói, con dê và cái bắp cải*):

Một con sói, một con dê và một cái bắp cải đang ở bờ sông. Người lái đò phải đưa chúng sang sông. Nhưng thuyền quá bé nên mỗi chuyến chỉ chở được một "hành khách" thôi. Vì những lý do mà ai cũng biết, không thể bỏ mặc sói với dê hoặc dê với bắp cải mà không có người trông. Vậy người lái đò phải xử trí thế nào mà vẫn đưa được sói, dê và bắp cải sang bên kia sông.

Xây dựng đồ thị vô hướng với các đỉnh thể hiện các hành khách còn lại bên phía xuất phát tại mỗi thời điểm khác nhau. Cạnh nối hai đỉnh thể hiện một chuyến đò qua sông.



Hình 8.1. Hành trình qua sông của sói, dê và bắp cải

Bài toán đưa về việc tìm đường đi ngắn nhất từ đỉnh a đến đỉnh b trên đồ thị. Đường đi như thế được chỉ ra bởi các mũi tên ở Hình 8.1.

8.2. BÀI TOÁN ĐƯỜNG ĐI CÓ TRỌNG SỐ BÉ NHẤT

Với bài toán đường đi tổng quát, xét các đồ thị có trọng số được định nghĩa như sau:

Định nghĩa 8.1:

Đồ thị G được gọi là đồ thị có trọng số nếu trên mỗi cạnh (i, j) của đồ thị được gán một số nguyên không âm $c(i, j)$.

Nhãn $c(i, j)$ trên cạnh (i, j) của đồ thị thường biểu diễn "chi phí" thực tế để đi qua cạnh này.

Ta thường ký hiệu đồ thị có trọng số là (G, c) .

Độ dài của đường đi trong đồ thị có trọng số bằng tổng các trọng số của các cạnh trên đường đi đó.

Bài toán: Cho đồ thị có trọng số (G, c) và hai đỉnh a, b thuộc G . Hãy tìm đường đi có trọng số bé nhất (nếu có) đi từ đỉnh a đến đỉnh b .

Độ dài đường đi ngắn nhất từ đỉnh a đến đỉnh b còn được gọi là khoảng cách từ đỉnh a đến đỉnh b trong đồ thị. Nếu không có đường đi từ a đến b thì đặt khoảng cách bằng ∞ .

8.3. THUẬT TOÁN DIJKSTRA TÌM ĐƯỜNG ĐI NGẮN NHẤT

Năm 1959, E. W. Dijkstra đưa ra một thuật toán rất hiệu quả để giải bài toán đường đi ngắn nhất.

Thuật toán thực hiện việc gán và giảm giá trị của nhãn $l(i)$ tại mỗi đỉnh i của đồ thị G như sau:

Thuật toán 8.2 (Tìm đường đi ngắn nhất):

1. Với đỉnh xuất phát a , gán nhãn $l(a) := 0$.
2. Nếu có cạnh (i, j) mà đỉnh i đã được gán nhãn và đỉnh j chưa được gán nhãn hoặc đỉnh j đã được gán nhãn nhưng
$$l(i) + c(i, j) < l(j)$$
 thì giảm nhãn $l(j) := l(i) + c(i, j)$.
3. Lặp lại bước 2 cho đến khi không gán hoặc giảm nhãn được nữa.

Định lý 8.1: Tại mỗi đỉnh b giá trị nhãn $l(b)$ cuối cùng (nếu có) chính là độ dài của đường đi ngắn nhất từ đỉnh a đến đỉnh b .

Chứng minh: Sau khi đã thực hiện xong thuật toán trên, nếu giá trị nhãn $l(b)$ xác định thì ta có đường đi từ đỉnh a tới đỉnh b .

Khởi phục đường đi từ a đến b như sau:

Xuất phát từ đỉnh b , tìm cạnh có đỉnh cuối là b và đỉnh đầu là i sao cho:

$$l(i) + c(i, b) = l(b).$$

Đỉnh i như thế chắc chắn phải tồn tại vì xảy ra đẳng thức ở lần gán hoặc giảm giá trị nhãn $l(b)$ cuối cùng. Cứ tiếp tục như thế cho đến khi gặp đỉnh a .

Giả sử ta nhận được dãy các cạnh:

$$(a, a_1), (a_1, a_2), \dots, (a_{k-1}, b)$$

mà trên đó:

$$l(a) + c(a, a_1) = l(a_1)$$

$$l(a_1) + c(a_1, a_2) = l(a_2)$$

.....

$$l(a_{k-1}) + c(a_{k-1}, b) = l(b).$$

Cộng từng vế và khử các giá trị chung ở cả hai vế ta có:

$$c(a, a_1) + c(a_1, a_2) + \dots + c(a_{k-1}, b) = l(b).$$

Vậy giá trị nhãn $l(b)$ chính là độ dài đường đi nói trên.

Bất kỳ đường đi nào khác từ đỉnh a đến đỉnh b cũng có các hệ thức tương tự nhưng có dấu $\geq$.

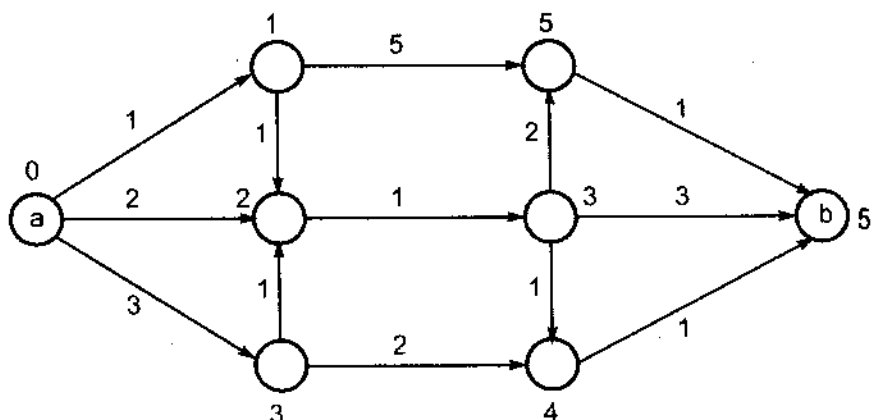
Vậy nhãn $l(b)$ là độ dài của đường đi ngắn nhất. ■

Ví dụ 8.2: Xét đồ thị có trọng số cho trong Hình 8.2.

Độ dài đường đi ngắn nhất từ đỉnh a đến đỉnh b là 5.

Để đơn giản việc tính toán, xây dựng ma trận trọng số C như sau:

$$C[i, j] = \begin{cases} c(i, j) & , \text{ nếu } (i, j) \in E \\ \infty & , \text{ nếu } (i, j) \notin E \\ 0 & , \text{ nếu } i = j. \end{cases}$$



Hình 8.2. Đồ thị có trọng số

Khi đó, thuật toán Dijkstra được trình bày chi tiết hơn như sau:

```

1  procedure DIJKSTRA(a) ;
2  begin
3    for  $j \in V$  do
4      begin
5         $L[j] := C[a, j]$  ;  $Truoc[j] := a$ 
6      end ;
7     $T := V \setminus \{a\}$  ;
8    while  $T \neq \emptyset$  do
9      begin
10       chọn đỉnh  $i \in T$  mà  $L[i] = \min\{L[j] \mid j \in T\}$  ;
11        $T := T \setminus \{i\}$  ;
12       for  $j \in T$  do
13         if  $L[j] > L[i] + C[i, j]$  then
14           begin
15              $L[j] := L[i] + C[i, j]$  ;
16              $Truoc[j] := i$  ;
17           end ;
18       end ;
19     end ;

```

Biến mảng $Truoc$ dùng để khôi phục đường đi.

8.4. ĐƯỜNG ĐI TRÊN ĐỒ THỊ PHI CHU TRÌNH

Sử dụng Thuật toán 4.1 (Chương 4) để đánh số các đỉnh trên đồ thị định hướng phi chu trình, xây dựng được thuật toán ngắn gọn hơn để tìm khoảng cách từ đỉnh nguồn tới tất cả các đỉnh trong một đồ thị phi chu trình.

Thuật toán 8.3:

Dữ liệu: Biểu diễn mảng DK_V các danh sách kề của đồ thị định hướng phi chu trình $G = (V, E)$ với tập đỉnh $V = \{v_1, v_2, \dots, v_n\}$ đã được đánh số mà danh sách $DK_V[v_j]$ chứa các đỉnh nhận v_j là đỉnh kề và ma trận trọng số C của đồ thị G .

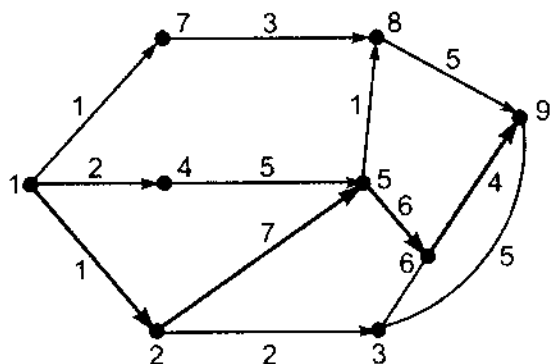
Kết quả: Mảng D các số nguyên với $D[v_i]$ chứa khoảng cách $d(v_1, v_i)$, $i = 2, 3, \dots, n$.

```
1  Begin
2   $D[v_1] := 0$  ;
3  for  $j := 2$  to  $n$  do  $D[v_j] := \infty$  ;
4  for  $j := 2$  to  $n$  do
5      for  $v_i \in DK\_V[v_j]$  do  $D[v_j] := \min(D[v_j],$ 
            $D[v_i] + C[v_i, v_j])$  ;
6  End.
```

Tính đúng đắn của thuật toán suy từ chi tiết sau đây: Tất cả các đỉnh trung gian trên đường đi ngắn nhất từ v_1 tới v_j có chỉ số nhỏ hơn j . Mỗi cạnh (v_i, v_j) được xét trong dòng lệnh 5 đúng một lần, do vậy độ phức tạp của thuật toán là $O(m)$.

Cũng có thể áp dụng thuật toán trên để tìm đường đi dài nhất từ đỉnh nguồn tới các đỉnh khác của đồ thị, hoặc tìm đường đi dài nhất trên đồ thị định hướng phi chu trình có trọng số.

Ví dụ 8.3: Tìm đường đi dài nhất trên đồ thị định hướng phi chu trình có trọng số dưới đây:



Hình 8.3. Đường đi dài nhất trên đồ thị phi chu trình có trọng số

8.5. ĐƯỜNG ĐI NGẮN NHẤT GIỮA TẤT CẢ CÁC CẶP ĐỈNH

Bài toán: Cho đồ thị có trọng số (G, c) . Hãy tìm đường đi ngắn nhất giữa tất cả các cặp đỉnh.

Bài toán này thường gặp trong việc xây dựng bảng khoảng cách giữa các thành phố, bảng giá cước vận chuyển giữa các nhà ga...

Bài toán này có thể giải quyết bằng cách sử dụng thuật toán Dijkstra với mỗi đỉnh của đồ thị lần lượt là các đỉnh xuất phát. Tuy nhiên, có thể giải quyết trực tiếp bài toán nhờ thuật toán Floyd như sau:

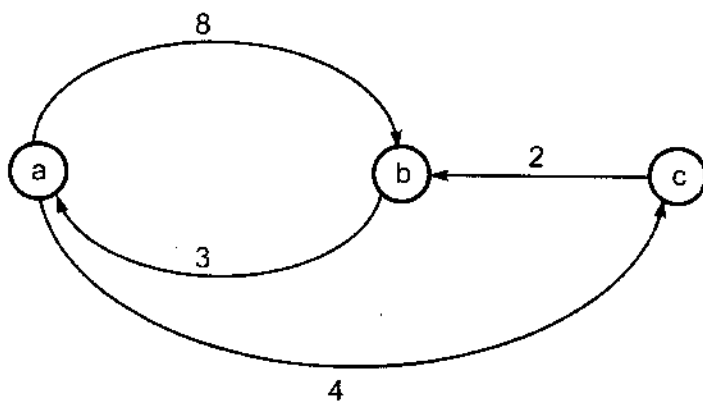
Sử dụng ma trận $D_n \times n$ để tính độ dài đường đi ngắn nhất giữa tất cả các cặp đỉnh.

1. Bắt đầu gán $D := C$ – ma trận trọng số.
2. Thực hiện n lần lặp trên D . Sau bước lặp thứ k , $D[i, j]$ chứa độ dài đường đi ngắn nhất từ đỉnh i đến đỉnh j mà chỉ đi qua các đỉnh có chỉ số không vượt quá k . Vậy trong bước lặp thứ k ta thực hiện theo công thức sau:

$$D^{(k)}[i, j] := \min (D^{(k-1)}[i, j], D^{(k-1)}[i, k] + D^{(k-1)}[k, j]),$$

với $k = 1, 2, \dots, n$.

Ví dụ 8.4: Giả sử ta có bản đồ giao thông sau:



Hình 8.4. Bản đồ giao thông

Các kết quả tính toán:

0	8	4
3	0	∞
∞	2	0

0	8	4
3	0	<u>7</u>
∞	2	0

0	8	4
3	0	7
<u>5</u>	2	0

0	<u>6</u>	4
3	0	7
5	2	0

Thuật toán 8.4 (Floyd):

Dữ liệu: Ma trận trọng số C của đồ thị.

Kết quả: Ma trận D cho biết khoảng cách của tất cả các cặp đỉnh.

```

1 BEGIN
2   for  $i := 1$  to  $n$  do
3     for  $j := 1$  to  $n$  do
4       begin  $D[i, j] := C[i, j]$ ;  $TRUOC[i, j] := 0$  end;
5     for  $k := 1$  to  $n$  do
6       for  $i := 1$  to  $n$  do
7         for  $j := 1$  to  $n$  do
8           if  $D[i, k] + D[k, j] < D[i, j]$  then

```



```

9      begin
10       $D[i, j] := D[i, k] + D[k, j];$ 
11       $TRUOC[i, j] := k$ 
12      end;
13 END.

```

Nếu $TRUOC[i, j] = 0$ thì đường đi ngắn nhất từ đỉnh i đến đỉnh j chính là cạnh (i, j) .

Để in ra các đỉnh trung gian trên đường đi ngắn nhất từ đỉnh i đến đỉnh j dùng thủ tục đệ quy sau:

```

1  procedure Duong_di( $i, j$ ) ;
2  begin
3     $k := TRUOC[i, j]$  ;
4    if  $k = 0$  then Exit ;
5    Duong_di( $i, k$ ) ;
6    write( $k$ ) ;
7    Duong_di( $k, j$ ) ;
8  end ;

```

Chẳng hạn, ma trận $TRUOC$ của ví dụ trên là:

0	3	0
0	0	1
2	0	0

Để xác định đường đi ngắn nhất từ đỉnh 1 đến đỉnh 2, lấy $k = TRUOC[1, 2] = 3$. Vậy đường đi ngắn nhất là: $\langle 1, 3, 2 \rangle$.

8.6. TÂM CỦA ĐỒ THỊ

Giả sử $G = (V, E)$ là một đồ thị có trọng số không âm trên các cạnh.

Ký hiệu $d(x, y)$ là khoảng cách giữa đỉnh x và đỉnh y trong đồ thị G .

Đại lượng $d(a) = \max \{ d(x, a) \mid x \in V \}$ được gọi là *độ lệch* của đỉnh a trong đồ thị G .

Các khái niệm bán kính, tâm và đường kính của một đồ thị được định nghĩa như sau:

Định nghĩa 8.2:

1. *Bán kính* R của đồ thị G là độ lệch bé nhất trên các đỉnh:

$$R = \min \{ d(a) \mid a \in V \}.$$

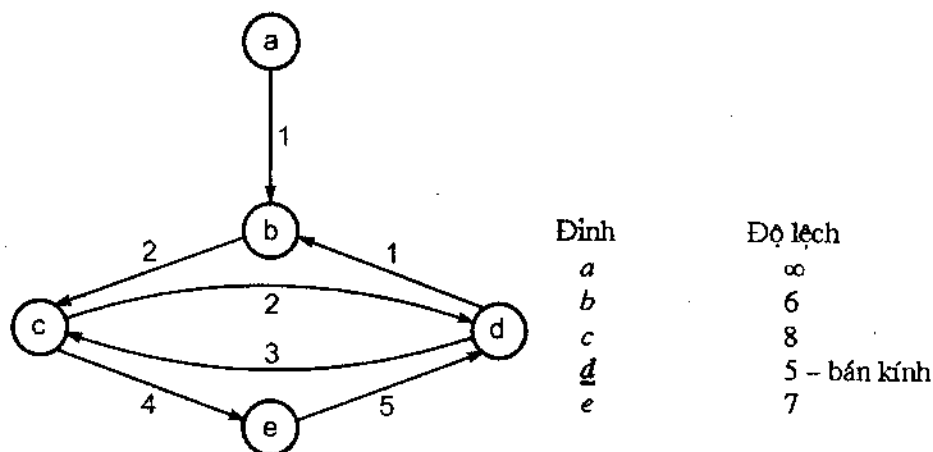
2. *Tâm* của đồ thị G là đỉnh a có độ lệch bé nhất:

$$\forall x \in V, d(a) \leq d(x).$$

3. *Đường kính* của đồ thị là khoảng cách dài nhất giữa các cặp đỉnh trong đồ thị:

$$d = \max \{ d(x, y) \mid x, y \in V \}.$$

Ví dụ 8.5: Cho đồ thị G như sau:



Hình 8.5. Đồ thị có trọng số

Vậy tâm của đồ thị là đỉnh d .

Ý nghĩa của tâm đồ thị: Dùng để xác định thủ đô của một nước, nút giao thông quan trọng trong một thành phố, vị trí đặt máy chủ trong một mạng máy tính...

Thuật toán 8.5 (Tìm tâm của đồ thị):

1) Dùng thuật toán Floyd để tính ma trận D các khoảng cách của các cặp đỉnh.

2) Tìm giá trị lớn nhất trên mỗi cột, cho ta độ lệch của đỉnh tương ứng.

3) Tìm đỉnh với độ lệch bé nhất, đó chính là tâm của đồ thị.

Ma trận khoảng cách D của Ví dụ 8.5 là:

a	b	c	d	e
0	1	3	5	7
∞	0	2	4	6
∞	3	0	2	4
∞	1	3	0	7
∞	6	8	5	0
∞	6	8	<u>5</u>	7
∞	6	8	5	7

Tâm: d

Bán kính: 5

Đường kính: ∞

Nếu vẽ một vòng tròn có tâm và bán kính như định nghĩa thì tất cả các đỉnh của đồ thị sẽ nằm trong vòng tròn này.

BÀI TẬP

- 8.1. Xây dựng đồ thị có hướng biểu diễn mạng lưới giao thông xe buýt ở Hà Nội, với các đỉnh là các điểm đỗ xe buýt, hai đỉnh được nối với nhau bằng một cạnh nếu có xe buýt đi từ đỉnh này tới đỉnh kia. Tìm đường đi ngắn nhất (*đường đi có số cạnh ít nhất*) từ nhà tới trường.
- 8.2. Chứng minh rằng, thuật toán duyệt đồ thị theo chiều rộng giúp giải được bài toán đường đi ngắn nhất, còn thuật toán duyệt theo chiều sâu thì không.

Chương 9

MẠNG VẬN TẢI

Khi điều hành một hệ thống vận tải, bao giờ chúng ta cũng mong muốn tìm ra một phương án vận chuyển được nhiều hàng hoá nhất. Chương này sẽ mô hình hoá toán học hệ thống vận tải và xây dựng thuật toán hữu hiệu chỉ ra phương án tối ưu ấy.

9.1. BÀI TOÁN LƯỠNG LỚN NHẤT

Bài toán luồng lớn nhất là một trong những bài toán tối ưu của Lý thuyết Đồ thị, được đề xuất vào đầu những năm 1950 và trở nên nổi tiếng với thuật toán Ford – Fulkerson.

9.1.1. Mạng vận tải

Định nghĩa 9.1: *Mạng vận tải là một đồ thị có hướng $G = (V, E)$ không có đỉnh nút, trong đó:*

- 1) Có duy nhất một đỉnh x_0 không có cạnh đi vào, $F^{-1}(x_0) = \emptyset$ (đỉnh phát);
- 2) Có duy nhất một đỉnh z không có cạnh đi ra, $F(z) = \emptyset$ (đỉnh thu);
- 3) Mỗi cạnh e được gán một số nguyên không âm $c(e)$ và gọi là khả năng thông qua của cạnh.

9.1.2. Luồng qua mạng

Với một mạng $G = (V, E)$, ký hiệu:

$W(x) = \{ (a, x) \in E \mid a \in V \}$ – đó là tập các cạnh đi vào đỉnh x .

$W^+(x) = \{ (x, b) \in E \mid b \in V \}$ – đó là tập các cạnh đi ra khỏi đỉnh x .

Định nghĩa 9.2:

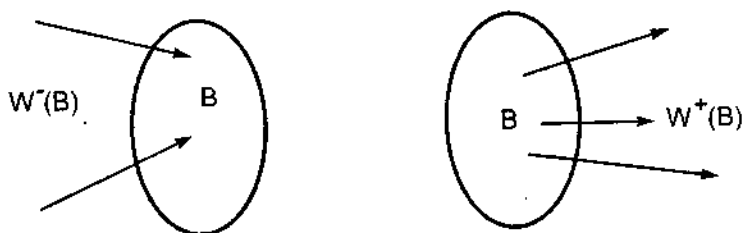
Hàm $t: E \rightarrow \mathbb{N}$ được gọi là một *luồng* đi qua mạng (G, c) nếu:

- a) $\forall e \in E: t(e) \leq c(e)$ – luồng trên mỗi cạnh không được vượt quá khả năng thông qua của cạnh đó.
- b) $\forall x \neq x_0$ và $z: t(W^-(x)) = t(W^+(x))$ – luồng trên các đỉnh phải cân bằng.

Với tập $B \subseteq V$, ký hiệu:

$W^-(B) = \{ (a, b) \in E \mid a \notin B, b \in B \}$ – tập cạnh từ ngoài B đi vào B ;

$W^+(B) = \{ (a, b) \in E \mid a \in B, b \notin B \}$ – tập cạnh từ B đi ra khỏi B .



Hình 9.1. Tập cạnh vào và ra của một tập đỉnh

Hiển nhiên, nếu tập con các đỉnh B không chứa x_0 và z thì:

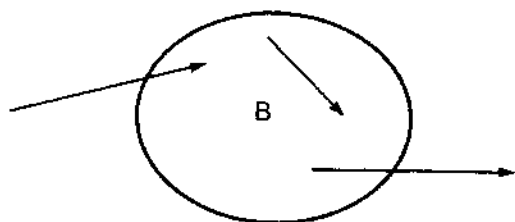
$$t(W^-(B)) = t(W^+(B)).$$

Thật vậy, theo tính chất b) của luồng:

$$\sum_{i \in B} t(W^-(x)) = \sum_{i \in B} t(W^+(x)).$$

Trong số các cạnh kể với đỉnh x , nếu có đỉnh đầu và đỉnh cuối đều nằm trong tập B thì nó sẽ có mặt ở cả hai vế của đẳng thức đúng một lần, do đó có thể giản ước. Sau khi giản ước, tổng ở vế trái chỉ còn lại các cạnh mà đỉnh đầu ở ngoài B , đỉnh cuối ở trong B , tức là

tập $W^-(B)$. Tương tự, tổng ở vế phải chỉ còn lại các cạnh mà đỉnh đầu ở trong B , đỉnh cuối ngoài B , tức là tập $W^+(B)$.



Hình 9.2. Các cạnh kề với một tập đỉnh

Từ nhận xét trên, nếu lấy $B = V \setminus \{x_0, z\}$ thì:

– khi G không có cạnh (x_0, z) ta có:

$$W^+(x_0) = W^-(B),$$

$$W^-(z) = W^+(B);$$

– và khi G có cạnh (x_0, z) thì:

$$W^+(x_0) = W^-(B) \cup (x_0, z)$$

$$W^-(z) = W^+(B) \cup (x_0, z)$$

Suy ra:

$$t(W^+(x_0)) = t(W^-(z)).$$

Ký hiệu: $tz = t(W^+(x_0))$ (cộng thêm $t(x_0, z)$, nếu có) và gọi là *giá trị của luồng qua mạng G* .

9.1.3. Bài toán luồng lớn nhất

Bài toán: Cho mạng vận tải (G, c) . Hãy tìm luồng t qua mạng sao cho tz đạt giá trị lớn nhất.

Để giải quyết bài toán này, dùng thuật toán Ford – Fulkerson.

9.1.4. Thuật toán Ford – Fulkerson tìm luồng lớn nhất

Đánh số đỉnh của mạng là: $0, 1, \dots, n$ sao cho đỉnh 0 là x_0 và đỉnh n là z .

Thuật toán 9.1 (Tìm luồng lớn nhất):

Ban đầu cho luồng $t = 0$ trên các cạnh.

Bước 1: *Đánh dấu các đỉnh của mạng.*

Lần lượt đánh dấu cho các đỉnh của mạng như sau:

- 1) Đỉnh x_0 được đánh dấu bằng số 0.
- 2) Nếu đỉnh x đã được đánh dấu, có cạnh (x, y) với đỉnh cuối y chưa được đánh dấu và $t(x, y) < c(x, y)$ thì đánh dấu cho đỉnh y là $+x$.
- 3) Nếu đỉnh y đã được đánh dấu, có cạnh (x, y) với đỉnh đầu x chưa được đánh dấu và $t(x, y) > 0$ thì đánh dấu đỉnh x là $-y$.
- 4) Nếu đỉnh thu z được đánh dấu là $+k$ với k nào đó thì có nghĩa là có một đường đi vô hướng từ x_0 đến z có dạng

$$\langle x_0, i_1, i_2, \dots, k, z \rangle,$$

trên đó mỗi đỉnh đã được đánh dấu $+j$ hoặc $-j$. Cụ thể là:

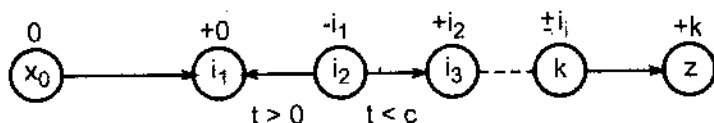
$$d(x_0) = 0$$

$$d(i_1) = +0$$

$$d(i_2) = \pm i_1$$

.....

$$d(z) = +k$$



Hình 9.3. Một đường đi vô hướng từ đỉnh phát đến đỉnh thu

Việc đánh dấu các đỉnh sẽ giúp khôi phục đường đi khi đã đến đỉnh z .

Bước 2: *Nâng giá trị của luồng.*

Xây dựng luồng mới t' như sau:

1) Nếu cạnh e không thuộc đường đi trên thì luồng giữ nguyên,
 $t'(e) := t(e)$.

2) Nếu cạnh e thuộc đường đi này và cùng chiều với chiều từ x_0 tới z , khi đó trên cạnh này $t(e) < c(e)$, thì ta đặt:

$$t'(e) := t(e) + 1.$$

3) Nếu cạnh e thuộc đường đi này và ngược chiều với chiều từ x_0 tới z , khi đó trên cạnh này $t(e) > 0$, thì ta đặt:

$$t'(e) := t(e) - 1.$$

Dễ thấy rằng, t' vẫn là một luồng và: $t'z = tz + 1$, nghĩa là ta đã nâng luồng thêm 1.

Lặp lại hai bước trên, chừng nào còn có thể để cải tiến luồng t .

Ta sẽ chứng minh rằng, khi thuật toán kết thúc thì luồng cuối cùng sẽ là luồng lớn nhất qua mạng. Trước hết ta có:

Bổ đề 9.1: Nếu t là một luồng qua mạng thì:

$$tz \leq \min \{ c(W^-(B)) \mid B \text{ chứa } z \text{ không chứa } x_0 \}.$$

Chứng minh: Giả sử B là một tập đỉnh chứa z nhưng không chứa x_0 . Xét luồng t :

Đặt: $B_1 = B \setminus \{z\}$, thế thì $B = B_1 \cup \{z\}$. Vì tập B_1 không chứa x_0 và z nên theo nhận xét ở trên, ta có:

$$t(W^-(B_1)) = t(W^+(B_1)).$$

$$W^-(B) = \{(a, b) \in E \mid a \notin B, b \in B\} = W^-(B_1) \cup W_1,$$

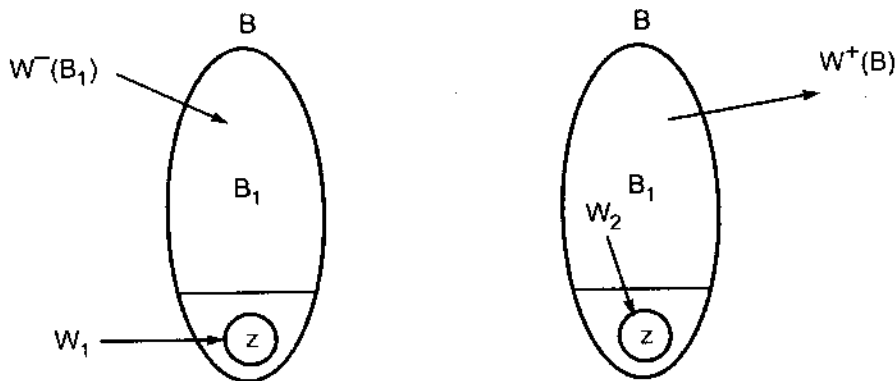
trong đó: $W_1 = \{(a, z) \in E \mid a \notin B\}$.

Hai tập $W^-(B_1)$ và W_1 rời nhau, nên:

$$t(W^-(B)) = t(W^-(B_1)) + t(W_1).$$

Vậy thì:

$$t(W^-(B_1)) = t(W^-(B)) - t(W_1) \quad (1)$$



Hình 9.4. Các tập cạnh vào và ra của B_1

Mặt khác,

$$W^+(B) = \{(a, b) \in E \mid a \in B, b \notin B\} = W^+(B_1) \setminus W_2$$

trong đó:

$$W_2 = \{(a, z) \in E \mid a \in B_1\}.$$

Vì $W_2 \subseteq W^+(B_1)$ cho nên:

$$t(W^+(B)) = t(W^+(B_1)) - t(W_2).$$

Thế thì

$$t(W^+(B_1)) = t(W^+(B)) + t(W_2) \quad (2)$$

Từ (1) và (2) suy ra:

$$t(W^-(B)) - t(W_1) = t(W^+(B)) + t(W_2).$$

Vì $W_1 \cap W_2 = \emptyset$ và $W_1 \cup W_2 = W^-(z)$ nên

$$t(W^-(B)) - t(W^+(B)) = t(W_1) + t(W_2) = tz.$$

Vậy với t là một luồng nào đó thì:

$$tz = t(W^-(B)) - t(W^+(B))$$

với mọi tập B chứa z và không chứa x_0 .

Do đó:

$$tz \leq t(W^-(B)) \leq c(W^-(B)).$$

Bổ đề được chứng minh. ■

Tập cạnh $W^-(B)$ được gọi là một *thiết diện* của mạng. Theo Bổ đề 9.1 thì giá trị của mọi luồng qua mạng không vượt quá khả năng thông qua của mọi thiết diện của mạng.

Ta trở lại chứng minh tính lớn nhất của luồng khi thuật toán dừng.

Định lý 9.1: Khi thuật toán Ford – Fulkerson dừng thì luồng cuối cùng nhận được sẽ là luồng lớn nhất với giá trị của luồng qua mạng là:

$$tz = \min \{c(W^-(B)) \mid B \text{ chứa } z \text{ và không chứa } x_0\}.$$

Chứng minh: Khi thuật toán dừng, có nghĩa là không đánh dấu được đến đỉnh z .

Ký hiệu B là tập các đỉnh không được đánh dấu.

Tập B không chứa đỉnh x_0 vì ta luôn đánh dấu từ x_0 , tập B chứa đỉnh z . Theo Bổ đề 9.1 thì:

$$tz = t(W^-(B)) - t(W^+(B)).$$

– Nếu cạnh $(a, b) \in W^-(B)$ thì a được đánh dấu (vì a nằm ngoài B) và b không được đánh dấu (b thuộc B). Điều đó có nghĩa là:

$$t((a, b)) = c((a, b)).$$

– Nếu cạnh $(a, b) \in W^+(B)$ thì b được đánh dấu (vì b nằm ngoài B) và a không được đánh dấu (a thuộc B). Do vậy, $t((a, b)) = 0$.

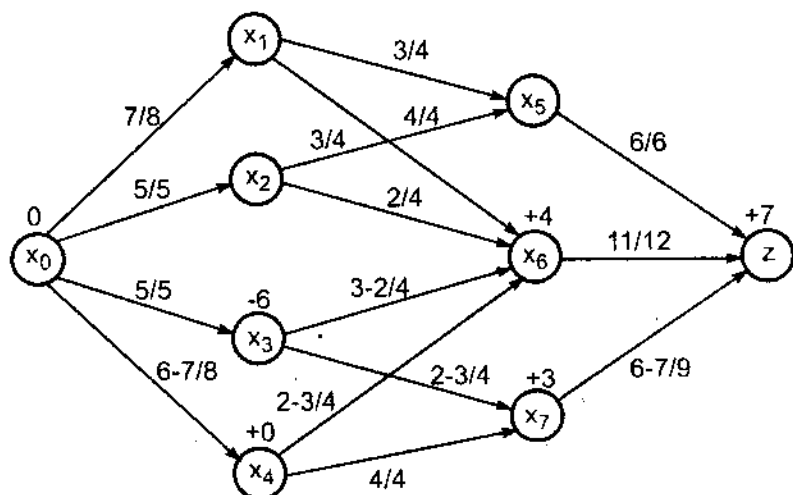
Vậy thì

$$tz = c(W^-(B)) - 0 = c(W^-(B)) \geq \min c(W^-(B)) \geq tz. \blacksquare$$

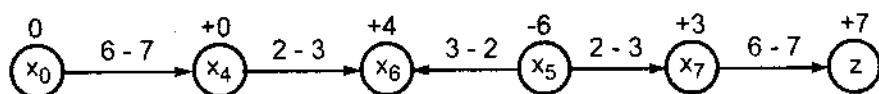
Tập cạnh $W^-(B)$ được gọi là *thiết diện hẹp nhất* của mạng.

Ví dụ 9.1: Xét mạng vận tải cho trong Hình 9.5.

Luồng hiện thời là $tz = 23$, ta sẽ nâng luồng này lên. Đánh dấu các đỉnh của mạng, nhận được đường đi vô hướng như Hình 9.6.



Hình 9.5. Mạng vận tải và một luồng trên mạng



Hình 9.6. Nâng luồng hiện thời

Tiến hành việc nâng luồng trên đường đi, nhận được luồng mới có giá trị $tz' = 24$. Tiếp tục nâng luồng sẽ nhận được luồng lớn nhất của mạng vận tải này là 26.

9.2. MỘT SỐ ỨNG DỤNG CỦA BÀI TOÁN LUỒNG LỚN NHẤT

Bài toán luồng lớn nhất có rất nhiều ứng dụng trong việc giải quyết các bài toán khác nhau của Lý thuyết Đồ thị.

9.2.1. Bài toán luồng nhỏ nhất

Ngược lại với bài toán luồng lớn nhất, chúng ta xét bài toán sau đây:

Bài toán: Cho mạng (G, c) . Tìm luồng t qua mạng có giá trị tz nhỏ nhất và thoả mãn điều kiện a') thay cho điều kiện a) (Định nghĩa 9.2) như sau:

$$a') \forall e \in E, t(e) \geq c(e).$$

Thuật toán 9.2 (Tìm luồng bé nhất):

Dùng phương pháp cải tiến luồng giống như phương pháp giải bài toán luồng lớn nhất.

Xuất phát từ một luồng t nào đó thoả mãn điều kiện c), ta dùng phương pháp sau đây để giảm giá trị của luồng t .

Bước 1: Đánh dấu các đỉnh

- 1) Đầu tiên đánh dấu cho đỉnh thu z số 0.
- 2) Nếu đỉnh y đã được đánh dấu, có cạnh (x, y) với đỉnh đầu chưa được đánh dấu và $t((x, y)) > c((x, y))$ thì đánh dấu cho đỉnh x là $+y$.
- 3) Nếu đỉnh x đã được đánh dấu, có cạnh (x, y) thì đánh dấu cho đỉnh y là $-x$.

Với cách đánh dấu này mà đi tới được đỉnh phát x_0 thì đã tìm được một đường đi vô hướng từ z tới x_0 được đánh dấu.

Bước 2: Giảm luồng

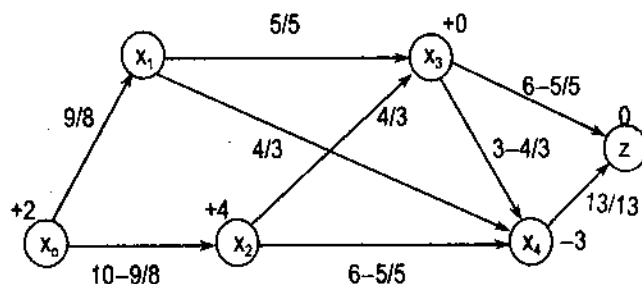
Bây giờ có thể giảm luồng đi 1 bằng cách chọn luồng mới t' như sau:

- 1) Nếu cạnh e không thuộc đường đi trên thì giữ nguyên luồng, nghĩa là:

$$t'(e) := t(e).$$

- 2) Nếu cạnh e thuộc đường đi này và cùng chiều với chiều từ x_0 tới z thì đặt $t'(e) := t(e) - 1$ (vì trên cạnh đó $t(e) > c(e)$), còn nếu cạnh e ngược chiều thì đặt: $t'(e) := t(e) + 1$.
- 3) Lặp lại quá trình giảm luồng trên cho đến khi không đánh dấu được tới đỉnh phát x_0 . Khi đó luồng nhận được có giá trị nhỏ nhất.

Ví dụ 9.2: Xét mạng vận tải sau:



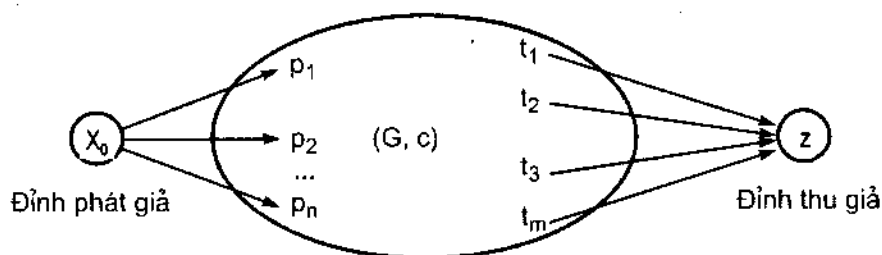
Hình 9.7. Mạng vận tải và luồng đã giảm

Luồng cũ có giá trị là $tz = 19$. Luồng mới sau khi cải tiến có giá trị là $tz' = 18$ và là luồng nhỏ nhất.

9.2.2. Bài toán luồng trên mạng có nhiều đỉnh phát và đỉnh thu

Giả sử (G, c) là một mạng vận tải với n đỉnh phát: $p_1, p_2, \dots, p_n$ và m đỉnh thu: $q_1, q_2, \dots, q_m$.

Bài toán tìm luồng lớn nhất từ nhiều đỉnh phát tới nhiều đỉnh thu có thể đưa về bài toán luồng lớn nhất từ một đỉnh phát tới một đỉnh thu bằng cách thêm vào một *đỉnh phát giả* X_0 , một *đỉnh thu giả* Z , các cạnh nối X_0 với tất cả các đỉnh phát và các cạnh nối tất cả các đỉnh thu với Z .



Hình 9.8. Mạng vận tải có nhiều đỉnh phát và nhiều đỉnh thu

Khả năng thông qua của các cạnh mới như sau:

- Nếu lượng phát của đỉnh p_i bị hạn chế bởi l_i thì đặt $c(X_0, p_i) = l_i$, còn nếu không bị hạn chế thì đặt bằng ∞ .

– Tương tự như thế, giới hạn của lượng thu của đỉnh t_j sẽ là khả năng thông qua của cạnh (t_j, Z) .

9.2.3. Bài toán tìm cặp ghép lớn nhất của đồ thị hai phần

Bài toán này là một dạng đặc biệt của bài toán mạng với nhiều đỉnh phát và nhiều đỉnh thu. Đưa bài toán này về bài toán luồng lớn nhất qua mạng.

Giả sử đồ thị $G = (V_1, V_2, F)$ là đồ thị hai phần. Xây dựng mạng vận tải như sau:

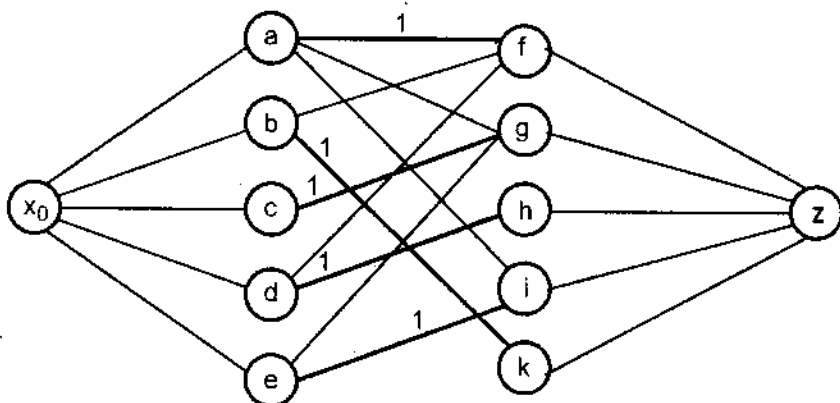
Các đỉnh của mạng là các đỉnh của đồ thị G và thêm vào đỉnh phát x_0 và đỉnh thu z . Mạng sẽ gồm tất cả các cạnh của G có hướng từ V_1 sang V_2 . Ngoài ra còn nối x_0 với tất cả các đỉnh trong V_1 và nối tất cả các đỉnh trong V_2 với z . Trên mọi cạnh e của mạng đều đặt $c(e) = 1$.

Khi đó mỗi luồng t qua mạng sẽ ứng với một cặp ghép W của G mà:

$$e \in W \Leftrightarrow t(e) = 1.$$

Ngược lại, mỗi cặp ghép W sẽ ứng với một luồng t qua mạng của G cũng theo quy tắc trên. Vậy t_z đạt giá trị lớn nhất khi W có nhiều cạnh nhất.

Ví dụ 9.3: Từ đồ thị hai phần có tập đỉnh $\{a, b, c, d, e, f, g, h, i, k\}$, xây dựng mạng vận tải như sau:



Hình 9.9. Mạng vận tải trên đồ thị hai phần

9.2.4. Bài toán vận tải với khả năng thông qua của các cạnh và các đỉnh

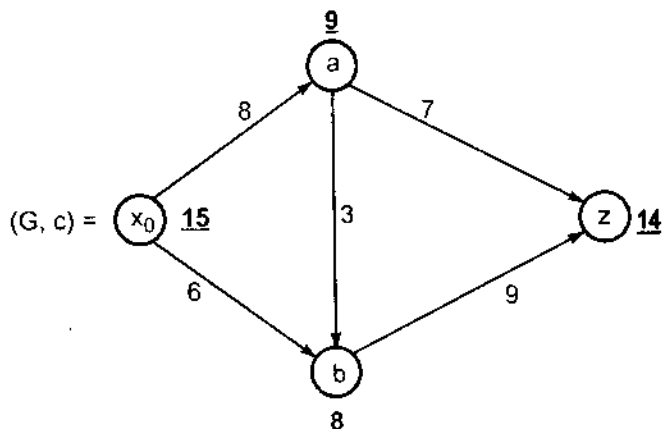
Giả sử trong đồ thị G , ngoài khả năng thông qua của các cạnh thì với mỗi đỉnh $x \in V$ còn có khả năng thông qua của đỉnh là $d(x)$ và đòi hỏi tổng luồng đi vào đỉnh x không được vượt quá $d(x)$, nghĩa là:

$$t(W^-(x)) \leq d(x).$$

Hãy tìm luồng lớn nhất giữa x_0 và z trong mạng này.

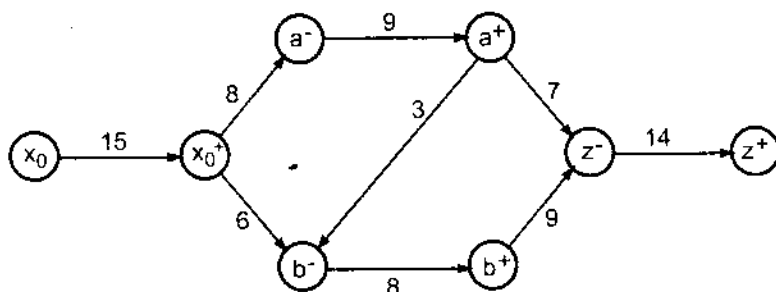
Để đưa bài toán này về bài toán luồng lớn nhất, xây dựng mạng G' sao cho: Mỗi đỉnh x trong G tương ứng với hai đỉnh x^- và x^+ trong G' , cạnh (x^-, x^+) thuộc G' và $c((x^-, x^+)) = d(x)$. Mỗi cạnh (x, y) trong G ứng với cạnh (x^+, y^-) trong G' .

Ví dụ 9.4: Xét mạng vận tải sau:



Hình 9.10. Mạng vận tải với khả năng thông qua cạnh và đỉnh

Xây dựng mạng (G', c) như sau:



Hình 9.11. Mạng vận tải tương ứng

Do luồng đi vào đỉnh x phải đi qua cạnh (x^-, x^*) với khả năng thông qua $d(x)$, nên luồng lớn nhất trong G' sẽ bằng luồng lớn nhất trong G và thoả mãn các điều kiện về khả năng thông qua của các cạnh và các đỉnh.

BÀI TẬP

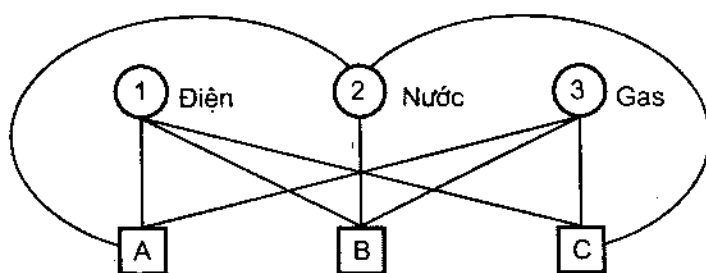
- 9.1. Cho một mạng vận tải và tìm luồng lớn nhất đi qua mạng này.
- 9.2. Cho một mạng vận tải với nhiều đỉnh phát và đỉnh thu, rồi tìm luồng lớn nhất đi qua mạng này.
- 9.3. Cho một đồ thị hai phần với các đỉnh bên trái chỉ các nhân viên, các đỉnh bên phải chỉ các công việc. Hãy áp dụng thuật toán tìm luồng lớn nhất để phân công công việc cho nhân viên (mỗi nhân viên đảm đương hai công việc).

Chương 10

ĐỒ THỊ PHẪNG

10.1. BÀI TOÁN BA BIỆT THỰ VÀ BA NHÀ MÁY

Trong một thị trấn có ba biệt thự và ba nhà máy cung cấp: điện, nước và khí đốt. Mỗi biệt thự đều muốn mắc đường cáp điện ngầm, đường ống cấp nước, đường ống cấp khí đốt riêng từ ba nhà máy đến nhà mình mà không gặp đường ống của các biệt thự khác. Hỏi rằng, có làm được những đường đi như thế hay không?



Hình 10.1. Ba biệt thự và ba nhà máy

Để giải quyết bài toán trên, sử dụng khái niệm đồ thị phẳng.

10.2. ĐỒ THỊ PHẪNG

Định nghĩa 10.1: Đa đồ thị vô hướng G được gọi là *đồ thị phẳng* nếu có thể biểu diễn nó trên mặt phẳng sao cho không có hai cạnh nào cắt nhau, trừ tại đỉnh.

Ví dụ, từ một bản đồ địa lý thế giới, xây dựng một đồ thị với mỗi nước là một đỉnh, hai đỉnh được nối với nhau bằng một cạnh nếu hai nước tương ứng có chung đường biên giới. Đồ thị nhận được là một đồ thị phẳng.

Giả sử G là một đồ thị phẳng được biểu diễn trên mặt phẳng.

Diện hữu hạn của một đồ thị phẳng là một miền kín của mặt phẳng, được giới hạn bằng các cạnh của đồ thị, sao cho có thể nối hai điểm bất kỳ thuộc diện này bằng một nét liền mà không gặp một cạnh nào ở bên trong.

Đồ thị còn có một *diện vô hạn*, đó là phần bù trên mặt phẳng của hợp các diện hữu hạn.

Ký hiệu h là số diện hữu hạn của một đồ thị phẳng.

Thấy rằng, hệ chu trình đơn độc lập cực đại sẽ chia đồ thị phẳng thành các diện hữu hạn. Thật vậy:

Định lý 10.1: Số diện hữu hạn của một đa đồ thị phẳng G bằng chu số của đồ thị này.

Chứng minh: Quy nạp theo số diện hữu hạn h của G .

$h = 1$: Chỉ có một chu trình đơn duy nhất, đó chính là biên của diện này. Suy ra chu số bằng 1.

$(h - 1) \Rightarrow (h)$: Giả sử đồ thị phẳng G với n đỉnh, m cạnh và p mảng liên thông có h diện. Lập đồ thị G' từ G bằng cách bớt đi cạnh e nào đó trên biên của một diện để số diện hữu hạn bớt đi 1. Khi đó, G' có $h - 1$ diện. Theo giả thiết quy nạp, chu số của G' là

$$h - 1 = (m - 1) - n + p$$

(p không đổi vì chỉ bớt đi một cạnh trên chu trình).

Suy ra số diện hữu hạn của G là $h = m - n + p =$ chu số của G . ■

Hệ quả 10.1: Nếu đa đồ thị phẳng G có n đỉnh, m cạnh, p mảng liên thông và h diện thì: $n - m + h = p + 1$ (công thức Euler tổng quát).

Chứng minh: Số diện của đồ thị phẳng bằng số diện hữu hạn cộng thêm 1 (diện vô hạn) = chu số + 1. Vậy thì, $h = m - n + p + 1$. Do đó, $n - m + h = p + 1$. ■

Hệ quả 10.2: Trong một đơn đồ thị phẳng có ít nhất một đỉnh có bậc không quá 5.

Chứng minh: Không mất tính tổng quát, có thể giả thiết đơn đồ thị là liên thông. Trong đơn đồ thị phẳng, mỗi diện hữu hạn được giới hạn bởi ít nhất 3 cạnh, mà mỗi cạnh thuộc nhiều nhất là hai diện, nên:

$$3h \leq 2m \Rightarrow h \leq \frac{2m}{3}.$$

Chứng minh bằng phản chứng: Giả sử mọi đỉnh của đồ thị G đều có bậc ít nhất bằng 6. Khi đó, tổng tất cả các bậc của các đỉnh trong $G = 2m \geq 6n$.

Do vậy: $m \geq 3n$, hay $n \leq \frac{m}{3}$.

Theo công thức Euler thì: $n - m + h = 1 + p = 2$.

Ta có: $2 \leq \frac{m}{3} - m + \frac{2m}{3} = 0$.

Suy ra điều vô lý. ■

10.3. CÁC ĐIỀU KIỆN CHO TÍNH PHẪNG CỦA ĐỒ THỊ

Với điều kiện nào đảm bảo cho một đồ thị là phẳng? Để trả lời cho câu hỏi này, dựa vào một số khái niệm được định nghĩa dưới đây. Trước hết, ta có ngay những kết quả hiển nhiên sau:

Định lý 10.2:

Giả sử G là một đồ thị và G' là đồ thị con của nó.

- 1) Đồ thị G phẳng thì G' cũng phẳng.
- 2) Đồ thị G' không phẳng thì G cũng không phẳng.

Chứng minh: Hiển nhiên. ■

Ký hiệu δ là độ dài của chu trình ngắn nhất, hoặc là số cạnh của đồ thị G nếu nó không có chu trình. Số δ được gọi là *đai* của đồ thị.

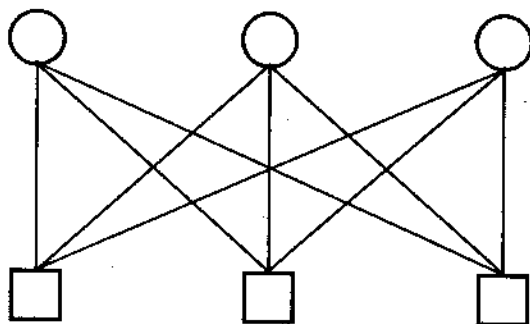
Định lý 10.3: Nếu đồ thị G là phẳng và đại của nó $\delta \geq 3$ thì

$$m \leq \frac{\delta}{\delta-2}(n-2).$$

Chứng minh: Ta có: $h.\delta \leq 2m$. Do vậy, theo công thức Euler thì $\delta(m-n+2) \leq 2m$.

Từ đó suy ra điều phải chứng minh. ■

Ví dụ 10.1: (Bài toán ba biệt thự và ba nhà máy)



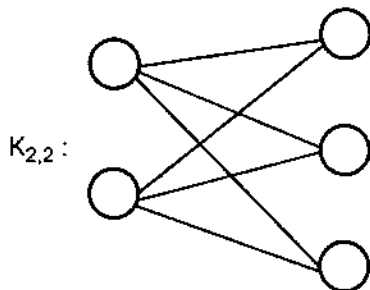
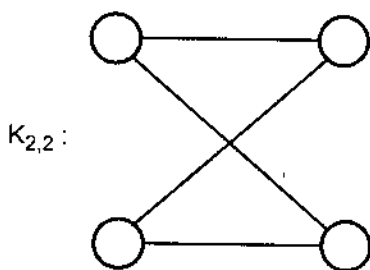
Hình 10.2. Đồ thị của bài toán ba biệt thự và ba nhà máy

Đại của đồ thị trên là $\delta = 4$. Vậy thì:

$$m = 9 > \frac{4}{4-2}(6-2) = 8.$$

Theo Định lý 10.3, đồ thị trên không phẳng.

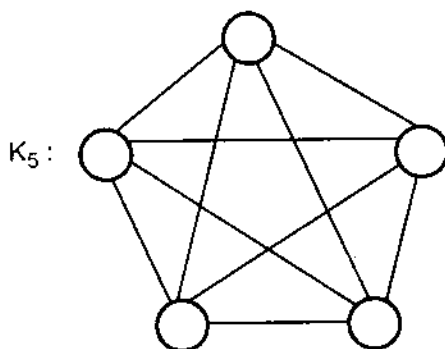
Ví dụ 10.2: Đồ thị hai phần đầy đủ $K_{m,n}$ là một đơn đồ thị có $m+n$ đỉnh, gồm m đỉnh "bên trái" và n đỉnh "bên phải", sao cho mỗi đỉnh "bên trái" đều kề với mọi đỉnh "bên phải".



Hình 10.3. Các đồ thị hai phần phẳng

Hệ quả 10.3: Đồ thị hai phân đầy đủ $K_{m,n}$ là đồ thị phẳng khi và chỉ khi $m \leq 2$ hoặc $n \leq 2$.

Ví dụ 10.3: Đồ thị đầy đủ 5 đỉnh (Hình 10.4).



Hình 10.4. Đồ thị đầy đủ 5 đỉnh

Đồ thị này có bậc $\delta = 3$. Vậy $m = 10 > \frac{3}{3-2}(5-2) = 9$. Do đó, đồ thị K_5 không phẳng. Từ đó suy ra, đồ thị đầy đủ K_n với $n \geq 5$ là không phẳng.

Chú ý rằng, đồ thị đầy đủ K_n với $n \leq 4$ là đồ thị phẳng.

Từ đồ thị G' cho trước, xây dựng đồ thị G bằng cách: Thêm vào G' các đỉnh mới và các cạnh mới. Đỉnh mới có thể nối với một đỉnh khác bằng một cạnh mới. Đỉnh mới cũng có thể đặt trên một cạnh cũ và chia cạnh này thành hai cạnh mới. Ta nói rằng, đồ thị G nhận được có chứa cấu hình G' . Hay đồ thị G' là một *cấu hình* của đồ thị G .

Chẳng hạn, đồ thị riêng của đồ thị là một cấu hình của đồ thị này.

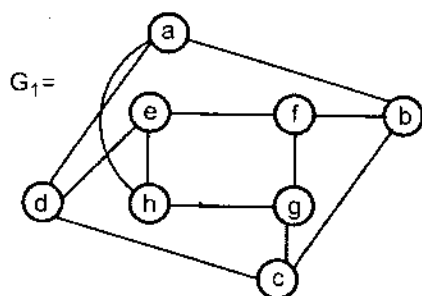
Định lý 10.4 (Kuratowski):

Đồ thị là phẳng khi và chỉ khi nó không chứa cấu hình $K_{3,3}$ hoặc K_5 .

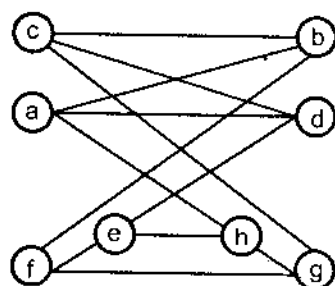
Có thể áp dụng định lý Kuratowski để xét tính chất phẳng của đồ thị.

Ví dụ 10.4: Xét các đồ thị sau:

1)



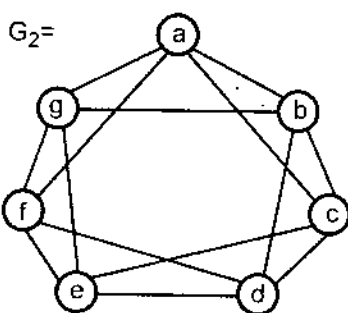
Vẽ lại :



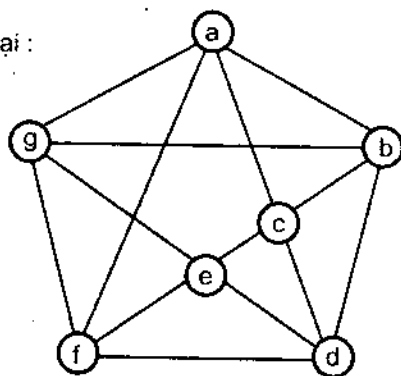
Hình 10.5. Hai đồ thị đẳng hình chứa cấu hình $K_{3,3}$

Đồ thị này chứa cấu hình $K_{3,3}$. Do vậy, nó không phẳng.

2)



Vẽ lại :



Hình 10.6. Hai đồ thị đẳng hình chứa cấu hình K_5

Đồ thị này chứa cấu hình K_5 . Vậy nó không phẳng.

10.4. SẮC SỐ CỦA ĐỒ THỊ PHẪNG

Do đặc thù của đồ thị phẳng, mà bài toán tô màu trên đồ thị phẳng trở nên rất lý thú.

Định lý 10.5 (Kempe – Heawood):

Mọi đồ thị phẳng không có đỉnh nút đều có sắc số không lớn hơn 5.

Chứng minh: Chứng minh định lý bằng quy nạp theo số đỉnh n của đồ thị.

$n = 1, 2, 3, 4, 5$: Hiển nhiên đúng.

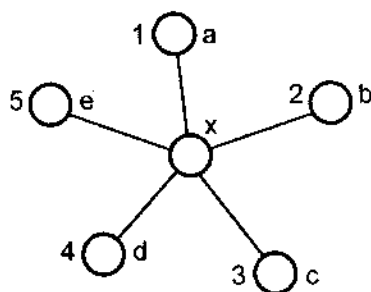
$(n - 1) \Rightarrow (n)$: Theo Hệ quả 10.2, đồ thị G có ít nhất một đỉnh x với bậc không quá 5. Xây dựng đồ thị G' từ đồ thị G bằng cách bỏ đỉnh x . Theo giả thiết quy nạp, đồ thị G' có sắc số không vượt quá 5.

Lấy một cách tô màu nào đấy của G' . Nếu các đỉnh kề với đỉnh x được tô bằng ít hơn 5 màu thì vẫn còn thừa màu để tô cho x . Sắc số của G bằng sắc số của G' (không vượt quá 5).

Vậy chỉ cần xét trường hợp đỉnh x kề với 5 đỉnh và các đỉnh kề với x được đánh số thứ tự theo chiều kim đồng hồ và tô bằng 5 màu như Hình 10.7.

Khi đó, phải đổi màu của các đỉnh trên để dành ra màu cho đỉnh x .

Xét tất cả các đường đi trong G bắt đầu từ đỉnh a , và gồm các đỉnh chỉ tô bằng màu 1 và màu 3. Trong các đường này, nếu không có đường nào đi qua đỉnh c thì có thể tráo đổi màu 1 với màu 3 cho tất cả các đỉnh trên các đường đi ấy. Sau đó, tô màu 1 cho đỉnh x .



Hình 10.7. Năm đỉnh kề với 5 màu

Ngược lại, nếu có một đường đi từ a đến c gồm toàn các đỉnh được tô bằng các màu 1 và màu 3, thì đường này cùng với hai cạnh (c, x) và (x, a) sẽ tạo thành một chu trình trong G . Do tính chất phẳng của đồ thị G , nên hai đỉnh b và d không thể cùng nằm bên trong hoặc cùng nằm bên ngoài chu trình này được. Suy ra không có đường đi nào nối b với d gồm các đỉnh chỉ tô bằng màu 2 và màu 4. Vậy lại có thể tráo đổi màu 2 với màu 4 cho tất cả các đỉnh trên các đường đi qua đỉnh b . Khi đó, hai đỉnh b và d có cùng màu 4. Tô màu 2 cho đỉnh x .

Định lý được chứng minh. ■

Bài toán bốn màu:

Bài toán bốn màu xuất hiện vào những năm 50 của thế kỷ XIX. Một thương gia người Anh tên là Gazri, khi tô màu bản đồ hành chính của nước mình, đã nhận ra rằng, luôn có thể tô được bằng 4 màu. Năm 1852, ông ta đã thông báo giả thuyết này cho De Morgan, và năm 1878 Keli đã đăng bài toán trên trong *Tuyển tập các công trình* của Hội Toán học Anh, gây nên sự chú ý của rất nhiều người.

Mãi đến tận năm 1976, ba nhà khoa học người Mỹ là K. Appel, W. Haken và J. Koch mới chứng minh bằng máy tính điện tử được rằng: giả thuyết của Gazri là đúng.

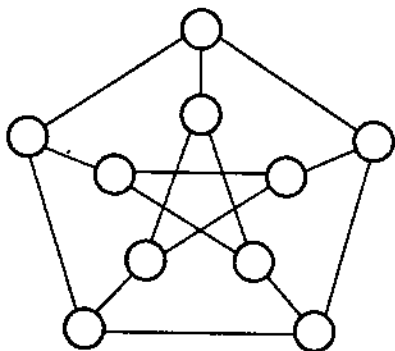
Định lý 10.6 (Appel – Haken):

Mọi đồ thị phẳng không có đỉnh nút đều có sắc số không quá 4.

Để thấy rằng, đồ thị vô hướng đầy đủ K_n ($n \geq 5$) có sắc số lớn hơn 4 nên không phẳng.

BÀI TẬP

- 10.1. Cho ba đồ thị. Hãy kiểm tra xem đồ thị nào là phẳng, đồ thị nào là không phẳng.
- 10.2. Với các đồ thị phẳng xây dựng được trong Bài tập 10.1, hãy tô màu và tìm sắc số của chúng.
- 10.3. Đồ thị Petersen dưới đây có phẳng hay không? Hãy giải thích.



Hình 10.8. Đồ thị Peterson 10 đỉnh

Chương 11

CÂY VÀ MỘT SỐ ỨNG DỤNG

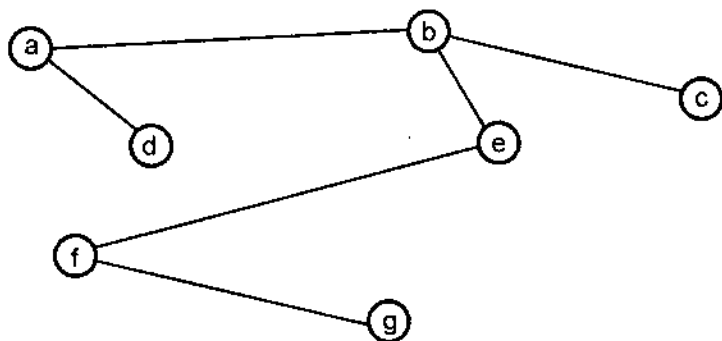
Trong chương này, xét một dạng đặc biệt nhưng có nhiều ứng dụng của đồ thị vô hướng, đó là khái niệm cây.

11.1. CÂY

Khái niệm cây được Cayley đưa ra đầu tiên vào năm 1857.

Định nghĩa 11.1: Giả sử $T = (V, E)$ là đồ thị vô hướng. Ta nói rằng đồ thị T là một *cây* nếu nó liên thông và không có chu trình.

Ví dụ 11.1: Đồ thị cho ở Hình 11.1 là một cây.



Hình 11.1. Cây 7 đỉnh

Kết quả dưới đây sẽ cho chúng ta một số tính chất lý thú và có thể dùng làm định nghĩa cho cây.

Định lý 11.1: Với đồ thị vô hướng T có số đỉnh không ít hơn 2, các tính chất sau đây là tương đương:

- 1) T là một cây.
- 2) T không có chu trình và có $n - 1$ cạnh.

- 3) T liên thông và có $n - 1$ cạnh.
- 4) T không có chu trình, nhưng nếu thêm một cạnh nối hai đỉnh bất kỳ không kề nhau thì xuất hiện một chu trình.
- 5) T liên thông, nhưng nếu bớt đi một cạnh bất kỳ thì sẽ mất tính liên thông.
- 6) Mỗi cặp đỉnh được nối với nhau bằng đúng một đường đi đơn.

Chứng minh: Chú ý rằng, đồ thị T không có chu trình khi và chỉ khi chu số của nó bằng 0, nghĩa là: $m = n - p$.

1) $\Rightarrow$ 2): Vì $p = 1$ và $m = n - p$, suy ra: $m = n - 1$.

2) $\Rightarrow$ 3): $m = n - p$, $m = n - 1$ cho nên $p = 1$.

3) $\Rightarrow$ 4): $p = 1$, $m = n - 1$, suy ra: $m = n - p$.

Vậy chu số của đồ thị $T = 0$, đồ thị T không có chu trình. Thêm một cạnh vào thì m tăng thêm 1; còn n , p không đổi. Khi đó chu số $c = m - n + p = 1$. Đồ thị có một chu trình.

4) $\Rightarrow$ 5): $c = 0$ nên $m = n - p$.

Giả sử ngược lại, đồ thị T không liên thông. Thế thì có ít nhất hai đỉnh a , b không liên thông. Khi thêm cạnh (a, b) vào, đồ thị vẫn không làm xuất hiện chu trình. Mâu thuẫn với điều 4).

Vậy đồ thị phải liên thông, nghĩa là $p = 1$. Suy ra: $m = n - 1$.

Khi bớt đi một cạnh bất kỳ, đồ thị vẫn không có chu trình. Do đó $m - 1 = n - p'$. Thế thì $p' = 2$ và đồ thị mất tính liên thông.

5) $\Rightarrow$ 6): Vì đồ thị T liên thông nên mỗi cặp đỉnh đều có đường đi đơn nối chúng. Giả sử cặp đỉnh a , b được nối bằng hai đường đi đơn khác nhau. Khi đó có cạnh e thuộc đường đi này, nhưng không thuộc đường đi kia. Bỏ cạnh e này đi, đồ thị vẫn liên thông. Trái với điều 5).

6) $\Rightarrow$ 1): Suy ra đồ thị T liên thông.

Giả sử T có chu trình. Vậy thì giữa hai đỉnh của chu trình có thể nối bằng hai đường đơn khác nhau. Mâu thuẫn với điều 6). ■

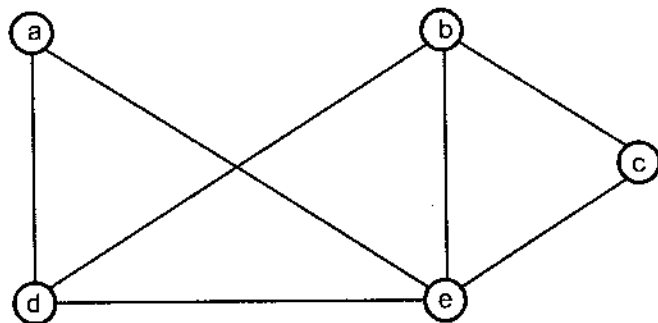
11.2. CÂY BAO TRÙM CỦA ĐỒ THỊ

Giả sử G là một đồ thị vô hướng.

11.2.1. Cây bao trùm

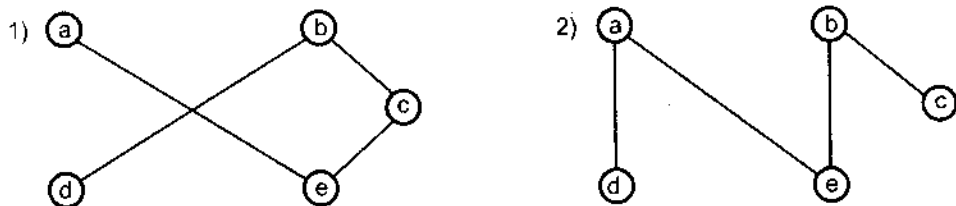
Định nghĩa 11.2: Cây T được gọi là *cây bao trùm* của đồ thị G nếu T là một đồ thị riêng của G .

Ví dụ 11.2: Đồ thị G được cho như sau:



Hình 11.2. Đồ thị có cây bao trùm

và một số cây bao trùm của G là:



Hình 11.3. Hai cây bao trùm của đồ thị Hình 11.2

Cây bao trùm có nhiều ứng dụng trong các bài toán điều khiển giao thông, nối các mạng điện...

Định lý 11.2: Đồ thị vô hướng G có cây bao trùm khi và chỉ khi G liên thông.

Chứng minh:

$\Rightarrow$: Hiển nhiên, vì cây bao trùm liên thông suy ra G liên thông.

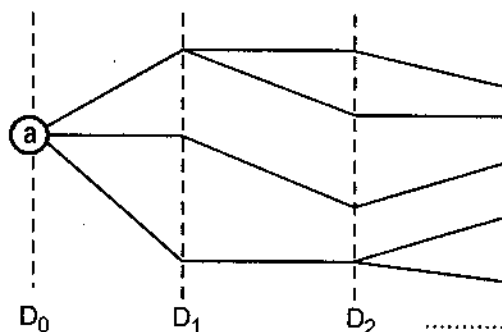
$\Leftarrow$: Chọn a là một đỉnh bất kỳ trong đồ thị G .

Ký hiệu $d(x)$ là độ dài của đường đi ngắn nhất nối đỉnh a với đỉnh x .

Lần lượt xây dựng các tập hợp

$$D_0 = \{a\}$$

$$D_i = \{x \mid d(x) = i\} \text{ với } i \geq 1.$$



Hình 11.4. Cách xây dựng cây bao trùm

Chú ý: Mỗi đỉnh x thuộc D_i ($i \geq 1$) đều có đỉnh y thuộc D_{i-1} sao cho (x, y) là một cạnh, vì nếu $\langle a, \dots, y, x \rangle$ là đường đi ngắn nhất nối a với x thì $\langle a, \dots, y \rangle$ là đường đi ngắn nhất nối a với y và $y \in D_{i-1}$.

Lập tập cạnh T như sau: Với mỗi đỉnh x của đồ thị G , thì $x \in D_i$ với $i \geq 1$, lấy một cạnh nào đó nối x với một đỉnh trong D_{i-1} . Tập cạnh này sẽ tạo nên một đồ thị riêng của G với n đỉnh và $n - 1$ cạnh. Đồ thị riêng này liên thông vì mỗi đỉnh đều được nối với đỉnh a . Theo tính chất 3) của cây thì T là một cây. Do vậy, T là cây bao trùm của đồ thị G . ■

Định lý 11.3 (Borchart): Số cây bao trùm của đồ thị vô hướng đầy đủ n đỉnh là n^{n-2} .

Kết quả trên cho thấy số lượng cây bao trùm của một đồ thị nói chung là rất lớn.

Các thuật toán duyệt đồ thị theo chiều rộng và chiều sâu là những công cụ tốt để tìm cây bao trùm của đồ thị liên thông.

Thuật toán 11.1 (Tìm cây bao trùm của đồ thị liên thông bằng phương pháp duyệt theo chiều sâu):

Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G .

Kết quả: Cây bao trùm (V, T) của đồ thị G .

```
1  procedure CBT_S ( $v$ ) ;
2  begin
3    Duyệt [ $v$ ] := true ;
4    for  $u \in \text{DK}[v]$  do
5      if ! Duyệt [ $u$ ] then
6        begin  $T := T \cup \{(v, u)\}$  ; CBT_S ( $u$ ) end ;
7    end ;
8  BEGIN    { Chương trình chính }
9    for  $u \in V$  do Duyệt [ $u$ ] := false ;
10    $T := \emptyset$  ;
11   CBT_S ( $z$ ) ; {  $z$  là đỉnh tùy ý của đồ
                   thị sẽ trở thành gốc của cây }
12 END.
```

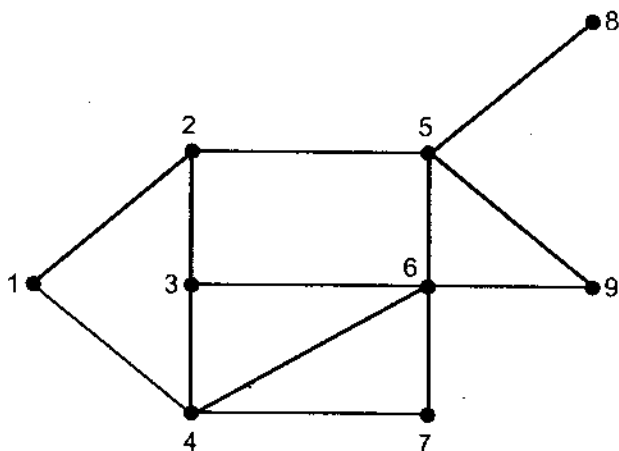
Tính đúng đắn của thuật toán được suy từ ba tính chất sau đây:

- 1) Khi thêm cạnh (v, u) vào tập cạnh T thì trong đồ thị (V, T) đã có đường đi từ z tới v . Vậy thì Thuật toán 11.1 xây dựng lên đồ thị liên thông.
- 2) Mỗi cạnh mới (v, u) được thêm vào tập T có đỉnh v đã được duyệt và đỉnh u đang duyệt. Vậy đồ thị đang được xây dựng không có chu trình.
- 3) Theo tính chất của phép duyệt theo chiều sâu, thủ tục CBT_S thăm tất cả các đỉnh của đồ thị liên thông G .

Do vậy, đồ thị do thuật toán xây dựng sẽ cho một cây bao trùm của đồ thị đã cho.

Hiển nhiên, độ phức tạp của thuật toán là $O(n + m)$.

Ví dụ 11.3: Áp dụng Thuật toán 11.1 cho đồ thị (nét mảnh), nhận được cây bao trùm (nét đậm) như Hình 11.5.



Hình 11.5. Cây bao trùm của đồ thị tìm theo phương pháp duyệt theo chiều sâu

Một cách tương tự, áp dụng phép duyệt đồ thị theo chiều rộng để tìm cây bao trùm của đồ thị liên thông và nhận được Thuật toán 11.2.

Thuật toán 11.2 (Tìm cây bao trùm của đồ thị liên thông bằng phương pháp duyệt theo chiều rộng):

Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G .

Kết quả: Cây bao trùm (V, T) của đồ thị G .

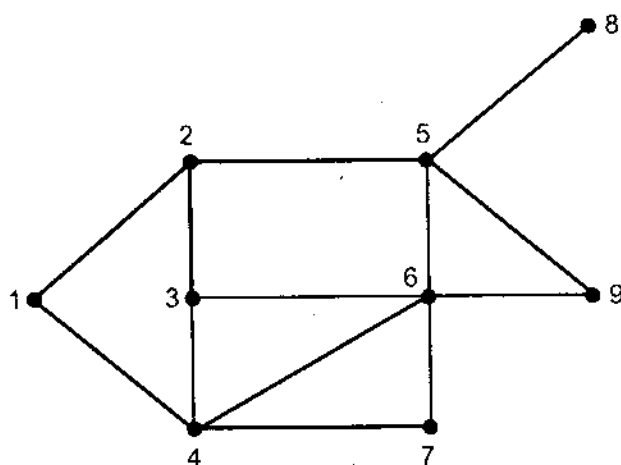
```
1 BEGIN
2   for  $u \in V$  do Duyệt  $[u] := \text{false}$  ;
3    $T := \emptyset$  ;
4    $Q := \emptyset$  ;
5   enqueue  $z$  into  $Q$  ; {  $z$  là đỉnh tùy ý của
                           đồ thị và là gốc của cây }
6   Duyệt  $[z] := \text{true}$  ;
7   while  $Q \neq \emptyset$  do
8     begin dequeue  $v$  from  $Q$  ;
```

```

9           for  $u \in DK[v]$  do
10              if ! Duyệt [u] then
11                 begin enqueue u into Q ;
12                     Duyệt [u] := true ;
13                      $T := T \cup \{(v, u)\}$ 
14             end
15 END.
```

Hiển nhiên, độ phức tạp của thuật toán là: $O(n + m)$.

Ví dụ 11.4: Áp dụng thuật toán trên cho đồ thị (nét mảnh) ta nhận được cây bao trùm (nét đậm) như sau:



Hình 11.6. Cây bao trùm của đồ thị tìm theo phương pháp duyệt theo chiều rộng

11.2.2. Một số ứng dụng của cây bao trùm

1) Kiểm tra tính liên thông của một đồ thị: Đồ thị là liên thông khi và chỉ khi nó có cây bao trùm.

2) Xây dựng hệ cơ sở của các chu trình.

Trước hết, giả thiết đồ thị liên thông $G = (V, E)$ có n đỉnh và m cạnh. Trong trường hợp đồ thị không liên thông thì xét từng thành phần liên thông.

Để xây dựng hệ cơ sở các chu trình thuộc G , tiến hành hai bước sau đây:

1. Xây dựng cây bao trùm T của G .

Giả sử trong quá trình xây dựng cây bao trùm T , đã bỏ đi các cạnh: $e_1, e_2, \dots, e_{m-n+1}$.

2. Xây dựng hệ chu trình cơ sở:

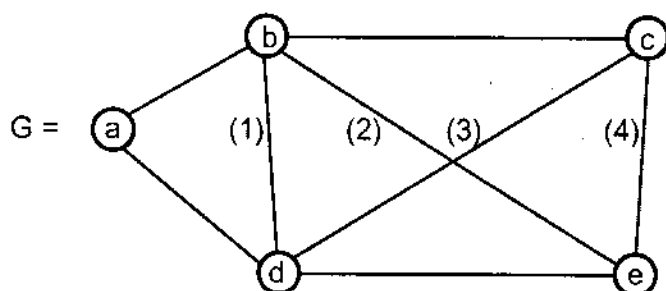
Lần lượt thêm vào cây T các cạnh e_i ($1 \leq i \leq m - n + 1$), nghĩa là khôi phục lại cạnh e_i trong G , khi đó sẽ xuất hiện chu trình α_i – đây cũng là chu trình của đồ thị G . Sau đó lại xoá cạnh e_i và thêm cạnh e_{i+1} vào. Ta nhận được các chu trình tương ứng: $\alpha_1, \alpha_2, \dots, \alpha_{m-n+1}$.

Hệ chu trình này độc lập vì: $\forall i \neq j$ thì α_i chứa e_i nhưng không chứa e_j , còn α_j chứa e_j nhưng không chứa e_i .

Hơn nữa, số các chu trình này là $m - n + 1 = m - n + p =$ chu số của $G =$ số các chu trình độc lập cực đại.

Vậy hệ chu trình tìm được là một cơ sở của các chu trình trong đồ thị G .

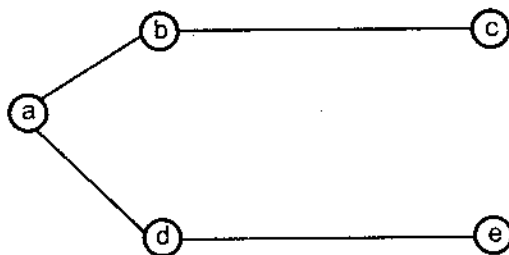
Ví dụ 11.5: Xét đồ thị vô hướng sau:



Hình 11.7. Đồ thị và các cạnh bỏ đi

$n = 5, m = 8, p = 1$. Vậy $c(G) = 4$.

Một cây bao trùm T của G là:



Hình 11.8. Một cây bao trùm của đồ thị Hình 11.7

Ta nhận được một hệ chu trình cơ sở:

$$\alpha_1 = [a, b, d]$$

$$\alpha_2 = [a, b, e, d]$$

$$\alpha_3 = [a, b, c, d]$$

$$\alpha_4 = [a, b, c, e, d].$$

11.3. CÂY BAO TRÙM NHỎ NHẤT

Bây giờ xét bài toán tổng quát tìm cây bao trùm.

11.3.1. Bài toán cây bao trùm nhỏ nhất

Cho đồ thị vô hướng G với tập cạnh E và hàm trọng số $c : E \rightarrow \mathbb{N}$. Hãy tìm cây bao trùm T của G sao cho tổng trọng số của các cạnh của T đạt giá trị nhỏ nhất.

Chẳng hạn như, xây dựng một hệ thống đường dây tải điện từ trạm phát điện đến các nơi tiêu thụ; nối các máy tính trong một mạng... sao cho dây điện sử dụng là ít nhất.

11.3.2. Các thuật toán tìm cây bao trùm nhỏ nhất

Giả sử G là một đồ thị vô hướng, liên thông và có trọng số. Khi đó, đồ thị G có cây bao trùm và sẽ có cây bao trùm nhỏ nhất.

Ta có thể dùng các thuật toán sau đây để tìm cây bao trùm nhỏ nhất của đồ thị G .

Thuật toán 11.3 (Thuật toán Kruskal):

- 1) Chọn cạnh có trọng số bé nhất, ký hiệu là e_1 và đặt $W := \{e_1\}$.
- 2) Giả sử đã chọn được $W = \{e_1, e_2, \dots, e_i\}$. Chọn e_{i+1} là cạnh có trọng số bé nhất trong số các cạnh còn lại trong $E \setminus W$ sao cho $\{e_1, e_2, \dots, e_i, e_{i+1}\}$ không chứa chu trình.
- 3) Đặt $W := W \cup \{e_{i+1}\}$.
- 4) Lập lại các bước 2) và 3) chừng nào còn có thể.

Tập cạnh W nhận được ở vòng lặp cuối cùng sẽ cho cây bao trùm nhỏ nhất.

Định lý 11.4: Tập các cạnh W tìm được theo thuật toán Kruskal là cây bao trùm nhỏ nhất của đồ thị G .

Chứng minh: Để việc chứng minh được đơn giản, ta xét hai trường hợp sau:

- 1) Trước hết xét trường hợp G là đồ thị vô hướng, đầy đủ, có trọng số.

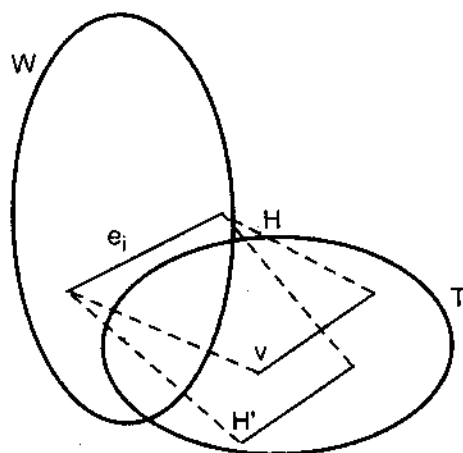
Tập cạnh W không có chu trình, nhưng nếu thêm một cạnh bất kỳ giữa hai đỉnh (cạnh này luôn có) sẽ tạo nên chu trình, vì nếu không có thì quá trình lặp chưa kết thúc. Do đó theo tính chất 4) của cây, tập cạnh W là một cây.

Mặt khác, mỗi đỉnh của đồ thị G đều kề với tập cạnh W , vì nếu không thì còn có thể thêm cạnh nữa vào W . Vậy trên W có n đỉnh và theo tính chất 3) của cây thì W có $n - 1$ cạnh. Suy ra cây W là cây bao trùm của đồ thị G .

Bây giờ sẽ chứng minh rằng, W là cây bao trùm nhỏ nhất của đồ thị G .

Giả sử T là một cây bao trùm nhỏ nhất nào đó của G . Ký hiệu e_i là cạnh đầu tiên của W không thuộc T , vậy thì:

$$\{e_1, e_2, \dots, e_{i-1}\} \subseteq T.$$



Hình 11.9. Cách thay cạnh của T với W

Theo tính chất 4) của cây, trong đồ thị $T \cup \{e_i\}$ có chu trình. Ký hiệu chu trình đó là H . Hiển nhiên, chu trình H phải chứa cạnh e_i . H cũng không thể nằm trọn trong tập cạnh W , vì trong W không có chu trình. Do vậy, có cạnh v trên chu trình này thuộc T nhưng không thuộc W .

Xét tập cạnh $T' = T \setminus \{v\} \cup \{e_i\}$.

– Tập cạnh T' không thể có chu trình. Vì nếu nó chứa chu trình H' nào đó thì H' phải chứa e_i và không chứa v . Khi đó, tập cạnh $(H' \setminus \{e_i\}) \cup (H \setminus \{e_i\})$ sẽ là một chu trình, và chu trình này phải nằm trong cây T , trái với tính chất 2) của cây.

– Số cạnh của T' là $n - 1$. Vậy theo tính chất 2) của cây thì tập cạnh T' là một cây và là cây bao trùm của đồ thị G .

Hơn nữa, vì T là cây bao trùm nhỏ nhất nên: $c(e_i) \geq c(v)$. Cạnh v không thể tạo với tập cạnh W ở bước lập $i - 1$ để có chu trình vì v nằm trong T . Nhưng nếu $c(e_i) > c(v)$ thì trong bước lập i ta đã không chọn cạnh e_i . Vậy thì: $c(e_i) = c(v)$ và trọng số của cây T bằng trọng số của cây T' . Nhận được cây bao trùm khác có trọng số không đổi, nhưng có thêm một cạnh chung với W là e_i .

Tiếp tục quá trình này sẽ nhận được cây bao trùm có trọng số bằng trọng số của cây T và trùng với W . Suy ra, cây W cũng là cây bao trùm nhỏ nhất.

2) Bây giờ xét trường hợp G là đồ thị vô hướng liên thông có trọng số.

Ký hiệu s là trọng số của đồ thị G .

Ta xây dựng đồ thị đầy đủ G' từ G như sau:

Nếu hai đỉnh trong G chưa kề nhau thì thêm một cạnh nối chúng với trọng số là $s + 1$. Khi đó, nếu T là một cây bao trùm của G thì cũng là một cây bao trùm của G' .

Áp dụng thuật toán Kruskal cho đồ thị đầy đủ G' , nhận được cây bao trùm nhỏ nhất W . Vậy thì, $s + 1 > \text{trọng số của } T \geq \text{trọng số của } W$ (vì W là cây bao trùm nhỏ nhất).

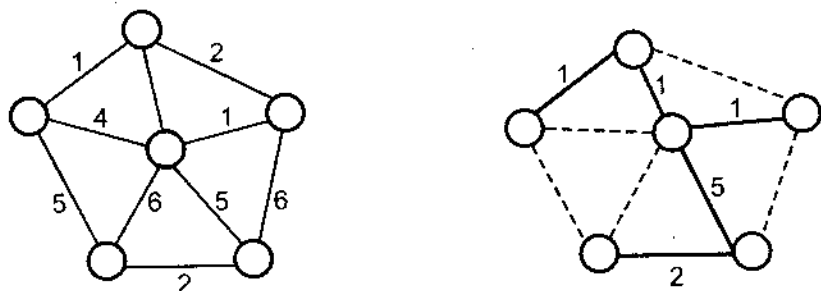
Vì trọng số của $W < s + 1$, suy ra cây W chỉ chứa các cạnh trong G . Thế thì, thuật toán Kruskal áp dụng cho đồ thị G' đã không hề chọn các cạnh mới thêm vào mà chỉ chọn các cạnh thuộc G .

Vậy cây W cũng là cây bao trùm nhỏ nhất của G . ■

Thuật toán Kruskal được mô tả chi tiết như sau:

```
1 procedure Kruskal ;
2 begin
3    $W := \emptyset$  ;    $Z := E$  ;
4   while ( $|W| < n - 1$ ) and ( $Z \neq \emptyset$ ) do
5     begin
6       chọn cạnh  $e$  có trọng số bé nhất trong  $Z$  ;
7        $Z := Z \setminus \{e\}$  ;
8       if  $W \cup \{e\}$  không chứa chu trình then
9          $W := W \cup \{e\}$  ;
9       end ;
10  if  $|W| < n - 1$  then
11    writeln(" Đồ thị không liên thông ")
11  end ;
```

Ví dụ 11.6: Hình 11.10 là đồ thị có trọng số và cây bao trùm nhỏ nhất của nó.



Hình 11.10. Đồ thị trọng số và một cây bao trùm nhỏ nhất

Thuật toán 11.4 (Thuật toán Prim):

Prim đã cải tiến thuật toán Kruskal như sau: Ở mỗi vòng lặp, chọn cạnh có trọng số bé nhất trong số các cạnh kề với các cạnh đã chọn mà không tạo nên chu trình.

Thuật toán Prim được gọi là *phương pháp lân cận gần nhất*. Theo phương pháp này, bắt đầu từ một đỉnh nào đó a của đồ thị G , nối nó với đỉnh "gần" nhất, chẳng hạn b . Nghĩa là, trong số các cạnh kề với a thì cạnh (a, b) được chọn có trọng số bé nhất. Tiếp theo, trong số các cạnh kề với đỉnh a hoặc đỉnh b , chọn cạnh có trọng số bé nhất mà không tạo nên chu trình với cạnh (a, b) . Cạnh này dẫn đến đỉnh thứ ba c ... Tiếp tục quá trình này cho đến khi nhận được cây gồm n đỉnh và $n - 1$ cạnh. Đó chính là cây bao trùm nhỏ nhất.

```

1 procedure Prim ;
2 begin
3   W := {cạnh có trọng số bé nhất } ;
4   for i := 1 to n - 2 do
5     begin
6       e := cạnh có trọng số bé nhất kề với cạnh
          trong W và nếu ghép nó vào W thì không tạo
          nên chu trình ;
7       W := W  $\cup$  {e}
8     end ;
9   end ;

```

Từ hai thuật toán đã trình bày ở trên, có thể khẳng định tính duy nhất của cây bao trùm nhỏ nhất trong trường hợp sau.

Định lý 11.5: Nếu đồ thị vô hướng liên thông G có trọng số trên các cạnh khác nhau từng đôi thì cây bao trùm nhỏ nhất tồn tại và duy nhất.

Chứng minh: Vì trong mỗi vòng lặp chỉ có duy nhất một cạnh được chọn. ■

11.4. CÂY BAO TRÙM LỚN NHẤT

Trong các thuật toán Kruskal và Prim không ràng buộc về dấu của trọng số, nên có thể áp dụng cho đồ thị vô hướng với trọng số trên các cạnh có cùng dấu tùy ý.

Vì vậy, để tìm cây bao trùm lớn nhất (tổng trọng số trên các cạnh của nó là lớn nhất) có hai cách sau đây:

1) Đổi thành dấu – cho các trọng số trên các cạnh. Áp dụng một trong hai thuật toán đã trình bày ở trên để tìm cây bao trùm nhỏ nhất. Sau đó đổi dấu + trở lại, sẽ được cây bao trùm lớn nhất.

2) Sửa đổi trong các thuật toán: Bước "*chọn cạnh có trọng số bé nhất...*" được thay bằng "*chọn cạnh có trọng số lớn nhất...*", còn các bước khác thì giữ nguyên. Khi thuật toán kết thúc, sẽ nhận được cây bao trùm lớn nhất.

11.5. CÂY PHÂN CẤP

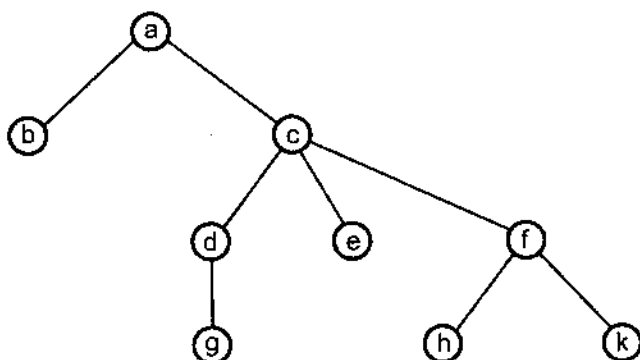
11.5.1. Định nghĩa cây phân cấp

Định nghĩa 11.3: *Cây phân cấp* là một cây, trong đó có một đỉnh đặc biệt gọi là gốc, giữa các đỉnh có mối quan hệ phân cấp "*cha-con*".

Số các con của một đỉnh trong cây phân cấp được gọi là *bậc* của đỉnh đó. Đỉnh không có con được gọi là *lá* của cây.

Thông thường, đỉnh không phải là lá được gọi là *đỉnh trong* của cây, còn lá được gọi là *đỉnh ngoài* của cây. Đỉnh gốc là đỉnh duy nhất không có cha.

Ví dụ 11.7: Cây T dưới đây có đỉnh gốc a , các đỉnh lá là b, g, e, h, k .



Hình 11.11. Cây phân cấp

– Mức của đỉnh trong cây phân cấp:

Gốc của cây có mức là 0.

Nếu mức của đỉnh cha là i thì mức của các đỉnh con là $i + 1$.

– Chiều cao của cây là mức cao nhất của các đỉnh trong cây.

Trong ví dụ trên, đỉnh gốc a có mức 0; các đỉnh b, c có mức 1; các đỉnh d, e, f có mức 2; các đỉnh g, h, k có mức 3. Cây có chiều cao là 3.

Cây phân cấp được áp dụng nhiều trong thực tế, chẳng hạn:

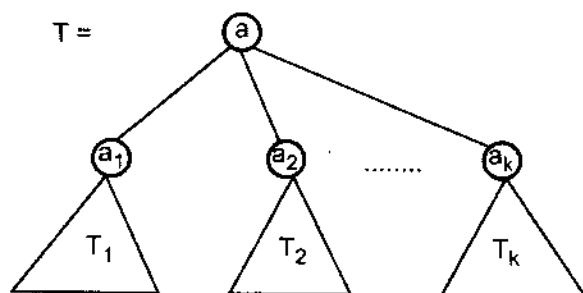
- 1) Mục lục của một cuốn sách để đọc giả tiện tra cứu.
- 2) Cấu trúc thư mục trên một ổ đĩa của máy tính để quản lý các tệp.
- 3) Sơ đồ tổ chức của một cơ quan để khách tiện liên hệ.

Để trình bày chặt chẽ các khái niệm khác và các phương pháp duyệt cây, ta đưa ra định nghĩa đệ quy cho cây phân cấp như sau:

Định nghĩa 11.4 (Đệ quy):

- 1) Tập rỗng là một cây phân cấp (cây rỗng).
- 2) Một đỉnh là một cây phân cấp.
- 3) Giả sử a là một đỉnh và $T_1, T_2, \dots, T_k$ là các cây phân cấp với các gốc là $a_1, a_2, \dots, a_k$ tương ứng. Cây T được xây dựng bằng

cách cho đỉnh a làm "cha" của các đỉnh $a_1, a_2, \dots, a_k$ sẽ là một cây phân cấp. Trong cây T này, đỉnh a là gốc và $T_1, T_2, \dots, T_k$ là các cây con của gốc a .



Hình 11.12. Cây phân cấp tổng quát

– Đường đi trong cây phân cấp T là một dãy các đỉnh $\langle b_1, b_2, \dots, b_m \rangle$ mà b_i là "cha" của b_{i+1} , $1 \leq i \leq m-1$.

Đường đi này đi từ đỉnh b_1 tới b_m trong cây T . Như vậy, đường đi trong cây phân cấp chỉ đi từ đỉnh "tổ tiên" xuống các đỉnh "con cháu".

Cây phân cấp T với bậc cao nhất của các đỉnh trong T là m , được gọi là cây m -phân.

Định lý 11.6: Giả sử T là một cây m -phân.

1. Nếu cây T có chiều cao h thì cây có nhiều nhất m^h lá.
2. Nếu cây T có l lá thì cây có chiều cao $h \geq \lceil \log_m l \rceil$.

Chứng minh:

1) Chứng minh quy nạp theo chiều cao h .

$h = 1$: Hiển nhiên.

$(h-1) \Rightarrow (h)$: Xét cây có chiều cao h . Bỏ gốc khỏi cây ta được một rừng gồm không quá m cây con, mỗi cây này có chiều cao $\leq h-1$. Theo giả thiết quy nạp, mỗi cây con có nhiều nhất m^{h-1} lá. Mà lá của những cây con này cũng là lá của cây T . Vậy cây T có nhiều nhất là $m \cdot m^{h-1} = m^h$ lá.

2) Vì số lá $l \leq m^h$ suy ra $h \geq \lceil \log_m l \rceil$. ■

11.5.2. Các cách duyệt cây

Duyệt cây là cách đưa ra một danh sách tuyến tính liệt kê tất cả các đỉnh của cây, mỗi đỉnh một lần.

Ba cách duyệt cây hay được dùng là:

- 1) Duyệt theo thứ tự trước (*pre-order search*).
- 2) Duyệt theo thứ tự giữa (*in-order search*).
- 3) Duyệt theo thứ tự sau (*post-order search*).

Các cách duyệt cây này được định nghĩa đệ quy như sau.

Định nghĩa 11.5:

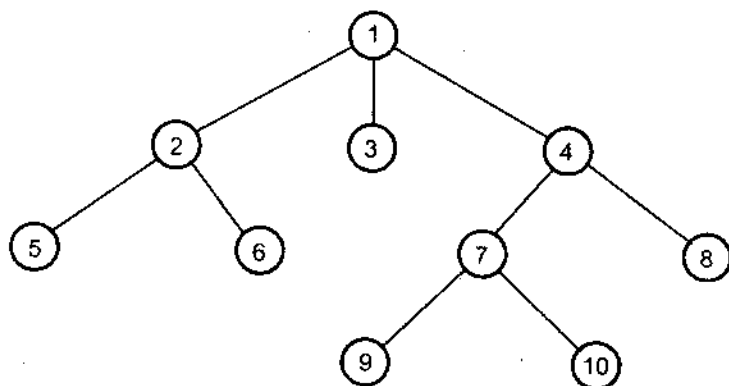
- Nếu cây T rỗng thì cả ba cách duyệt đều cho danh sách rỗng.
- Nếu cây T chỉ có một đỉnh thì cả ba cách duyệt đều cho danh sách gồm chỉ một đỉnh này.
- Nếu cây T có gốc a và các cây con $T_1, T_2, \dots, T_k$ thì:

1) *Duyệt theo thứ tự trước* của cây T là danh sách bao gồm gốc a , sau đó là các đỉnh của cây con T_1 được duyệt theo thứ tự trước, rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự trước... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự trước.

2) *Duyệt theo thứ tự giữa* của cây T là danh sách bao gồm các đỉnh của cây con T_1 được duyệt theo thứ tự giữa, sau đó là gốc a , rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự giữa... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự giữa.

3) *Duyệt theo thứ tự sau* của cây T là danh sách bao gồm các đỉnh của cây con T_1 được duyệt theo thứ tự sau, rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự sau... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự sau, và cuối cùng là đỉnh gốc a .

Ví dụ 11.8: Cây phân cấp T được cho như sau:



Hình 11.13. Cây phân cấp và kết quả của 3 cách duyệt

Duyệt theo thứ tự trước: 1, 2, 5, 6, 3, 4, 7, 9, 10, 8.

Duyệt theo thứ tự giữa: 5, 2, 6, 1, 3, 9, 7, 10, 4, 8.

Duyệt theo thứ tự sau: 5, 6, 2, 3, 9, 10, 7, 8, 4, 1.

Thủ tục duyệt một cây theo thứ tự giữa được mô tả như sau:

```
1 procedure D_GIUA (a) ;  
2   begin  
3     if a là lá then write(a)  
4     else  
5       begin  
6         D_GIUA (con bên trái nhất của a) ;  
7         write(a) ;  
8         for mỗi một con c của a, trừ con  
           bên trái nhất, từ trái sang phải do  
9           D_GIUA (c) ;  
10    end ;  
11  end ;
```

11.6. CÂY NHỊ PHÂN

Cây nhị phân là cây phân cấp mà mỗi đỉnh của nó có không quá hai con.

Các cây con của một đỉnh của cây nhị phân được phân biệt là cây con trái và cây con phải.

Định lý 11.7: Số các cây nhị phân n đỉnh là $c_n = \frac{1}{n+1} \binom{2n}{n}$.

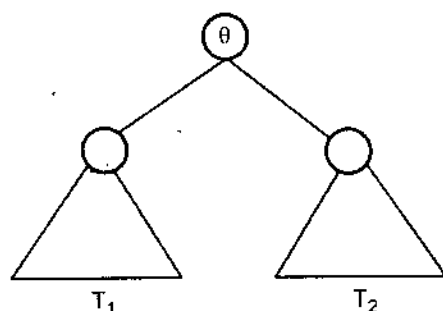
Dãy số c_n , $n = 0, 1, 2, \dots$ do Catalan tìm ra và được gọi là *dãy số Catalan*.

Cây nhị phân có rất nhiều ứng dụng trong thực tế.

11.6.1. Cây biểu thức

Cây biểu thức là cây nhị phân mà mỗi đỉnh của nó được gán nhãn theo quy tắc sau: Các lá được gán các đại lượng, còn các đỉnh trong được gán các dấu phép toán của một biểu thức nào đó.

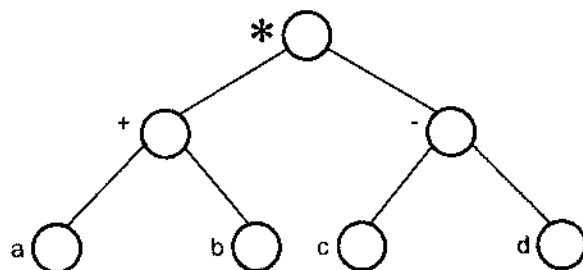
Cụ thể, nếu biểu thức $E = (E_1) \theta (E_2)$ và các cây biểu thức T_1, T_2 biểu diễn các biểu thức con E_1, E_2 thì cây biểu thức T biểu diễn E được xây dựng như Hình 11.14.



Hình 11.14. Cây biểu thức tổng quát

Ví dụ 11.9: Xét biểu thức: $E = (a + b) * (c - d)$.

Cây biểu thức tương ứng:



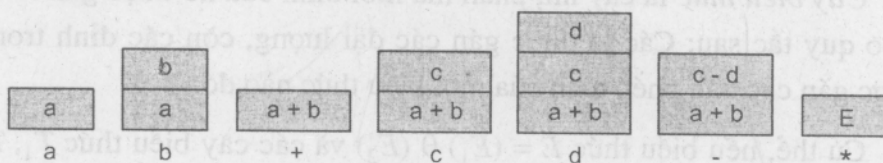
Hình 11.15. Cây biểu thức E

Duyệt cây biểu thức trên theo thứ tự sau, ta được danh sách

$$a b + c d - *$$

Đó là dạng Balan ngược của biểu thức E , giúp máy tính tính giá trị của biểu thức E một cách thuận tiện nhờ một stack lưu giữ các đại lượng.

Mỗi khi đọc một phép toán, máy thực hiện phép toán này với các đại lượng lấy ra từ đỉnh của stack, kết quả tính được lại đặt vào đỉnh của stack này. Cứ như thế cho đến khi thực hiện xong phép toán cuối cùng.



Hình 11.16. Dãy các stack phục vụ tính toán một biểu thức

11.6.2. Cây mã tiền tố

Cho một tập các ký hiệu. Hãy mã các ký hiệu này bằng dãy các chữ số 0, 1 thoả mãn tính chất tiền tố, nghĩa là không có mã của ký hiệu nào lại là tiền tố của mã của ký hiệu khác.

Để thực hiện công việc này, xây dựng một cây nhị phân sao cho mỗi ký hiệu tương ứng với một lá, cạnh đi xuống con trái của một đỉnh được gán nhãn 0, còn cạnh đi xuống con phải được gán nhãn 1.

Khi đó, dãy các nhãn trên đường đi từ gốc đến lá sẽ cho mã tiền tố của ký hiệu tương ứng.

Cây nhị phân xây dựng như trên được gọi là *cây mã tiền tố*.

Ví dụ 11.10:



Hình 11.17. Các cây mã tiền tố

Các bộ mã tiền tố nhận được là:

Ký hiệu	Bộ mã 1	Bộ mã 2
<i>a</i>	000	000
<i>b</i>	001	11
<i>c</i>	010	01
<i>d</i>	011	001
<i>e</i>	100	10

11.6.3. Cây mã Huffman

Giả sử có một bản tin là dãy các ký hiệu lấy trong một tập ký hiệu hữu hạn A . Biết rằng mỗi ký hiệu xuất hiện trong bản tin theo một tần suất đã biết. Hãy xây dựng bộ mã tiền tố cho tập A sao cho độ dài chuỗi mã của bản tin là ngắn nhất.

Bộ mã tìm được mang tính tối ưu và được gọi là *bộ mã Huffman*.

Gọi d là số ký hiệu của bản tin, $\varphi(x)$ là tần suất xuất hiện của ký hiệu x trong bản tin. Mỗi cây nhị phân T với nhãn 0, 1 trên các cạnh và có số lá bằng số ký hiệu của tập A sẽ cho một bộ mã tiền tố cho tập ký hiệu A . Mức $\mu(x)$ của lá x chính là chiều dài mã của ký hiệu x . Khi đó, độ dài chuỗi mã của toàn bộ bản tin sẽ là:

$$M = d \sum_{x \in A} \mu(x) \varphi(x).$$

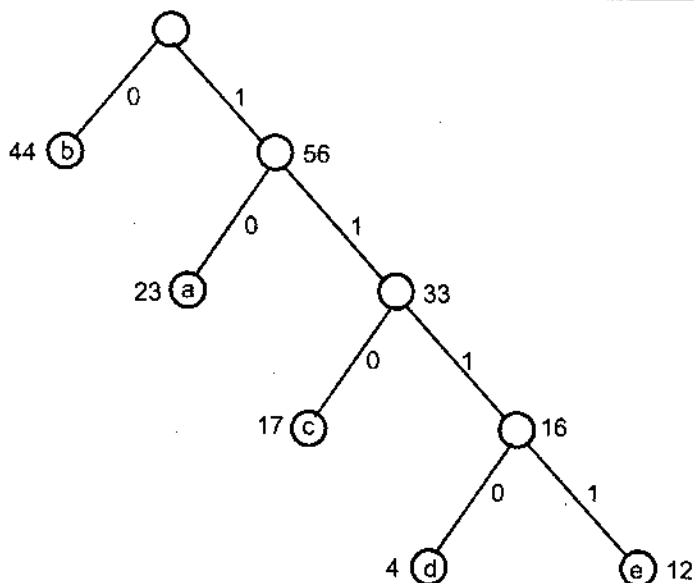
Cây mã tiền tố T là tối ưu khi độ dài M của mã bản tin đạt giá trị nhỏ nhất.

Ví dụ 11.11: Xét bản tin gồm 1000 ký hiệu trong tập ký hiệu $A = \{a, b, c, d, e\}$ với tần suất xuất hiện của các ký hiệu trong bản tin như sau:

Ký hiệu	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
Tần suất (%)	23	44	17	4	12

Bộ mã 1 và bộ mã 2 chọn giống như trong Ví dụ 11.10. Bộ mã 3 được xây dựng dựa trên cây mã tiền tố (tối ưu) ở Hình 11.18. Độ dài của mã bản tin tương ứng như sau:

Ký hiệu	Bộ mã 1	Bộ mã 2	Bộ mã 3
<i>a</i>	000	000	10
<i>b</i>	001	11	0
<i>c</i>	010	01	110
<i>d</i>	011	001	1110
<i>e</i>	100	10	1111
Độ dài mã bản tin:	3000	2270	2050



Hình 11.18. Cây mã tiền tố tối ưu

Trong cây mã tiền tố ở Hình 11.18, chúng ta đã xây dựng dựa trên nguyên lý: đỉnh lá nào có tần suất càng lớn thì đường đi từ gốc cây tới đỉnh lá đó càng ngắn.

Việc xây dựng bộ mã Huffman cho tập ký hiệu *A* được thực hiện thông qua xây dựng cây mã tiền tố tối ưu *T*. Thuật toán Huffman sẽ giúp xây dựng cây này.

Giả sử tập *A* có *l* ký hiệu.

Thuật toán 11.5

- 1) Xây dựng rừng T có l cây, mỗi cây chỉ gồm một đỉnh tương ứng với một ký hiệu x trong A và được gán nhãn $\varphi(x)$.
- 2) Chọn hai cây trong T có gốc với nhãn nhỏ nhất. Thêm một đỉnh mới với nhãn là tổng các nhãn của hai gốc cây vừa chọn. Nối đỉnh mới với hai gốc này bằng hai cạnh có nhãn 0, 1 để tạo thành một cây nhị phân.
- 3) Nếu T vẫn chưa phải là một cây thì lặp lại bước 2), ngược lại thì dừng.

Cây mã tiền tố trong Hình 11.18 được xây dựng dựa trên thuật toán này. Tính đúng đắn của Thuật toán Huffman được khẳng định qua định lý sau:

Định lý 11.8: Khi thuật toán Huffman dừng thì cây mã tiền tố nhận được là tối ưu.

Chứng minh: Chứng minh bằng quy nạp theo số ký hiệu l của tập A .

$l = 2$: Hiển nhiên.

$(l) \Rightarrow (l + 1)$: Giả sử thuật toán đúng với các tập có l ký hiệu, phải chứng minh nó cũng đúng với tập A có $l + 1$ ký hiệu.

Giả sử tập $A = \{x_1, x_2, \dots, x_{l-2}, x_{l-1}, x_l, x_{l+1}\}$. Không mất tính tổng quát, có thể giả thiết x_l, x_{l+1} có tần suất xuất hiện nhỏ nhất.

Ký hiệu H là cây mã tiền tố cho tập A theo thuật toán Huffman. Vì x_l, x_{l+1} có tần suất nhỏ nhất nên đã được chọn đầu tiên ở bước 2) và được thêm đỉnh mới y có tần suất là $\varphi(x_l) + \varphi(x_{l+1})$. Hiển nhiên, theo cách xây dựng của thuật toán thì cây nhị phân $H' = H \setminus \{x_l, x_{l+1}\}$ là cây mã tiền tố của tập $A' = \{x_1, x_2, \dots, x_{l-2}, x_{l-1}, y\}$ có l ký hiệu.

Theo giả thiết quy nạp, cây H' là cây mã tối ưu cho A' . Độ dài của mã bản tin theo cây H là:

$$M = d \sum_{x_i \in A} \mu(x_i) \varphi(x_i) = d \left[\sum_{i=1}^{l-1} \mu(x_i) \varphi(x_i) + \mu(y) \varphi(y) \right] + d(\varphi(x_l) + \varphi(x_{l+1}))$$

đạt giá trị bé nhất. Cây mã tiền tố H là tối ưu. ■

Cây mã tiền tố tối ưu và thuật toán Huffman được ứng dụng rộng rãi trong mật mã học.

BÀI TẬP

- 11.1.** Các phát biểu sau đây đúng hay sai? Hãy giải thích.
- 1) Nếu đồ thị có n đỉnh và $n - 1$ cạnh thì đó là một cây.
 - 2) Nếu đồ thị liên thông thì không có chu trình.
 - 3) Nếu bỏ bất kỳ cạnh nào của đồ thị liên thông cũng làm mất tính liên thông thì đồ thị đó không có chu trình.
 - 4) Nếu thêm một cạnh vào một cây thì sẽ sinh ra đúng một chu trình.
- 11.2.** Đồ thị G là một rừng gồm q cây và n đỉnh. Hãy tìm số cạnh của rừng G .
- 11.3.** Cho ba cây có ít nhất 15 đỉnh. Duyệt các cây này theo cả ba thứ tự: trước, giữa và sau.
- 11.4.** Chứng minh rằng, nếu đồ thị G có duy nhất một cây bao trùm thì G là một cây.
- 11.5.** Giả sử (x, y) là cạnh có trọng số bé nhất trong đồ thị G . Hãy chỉ ra rằng, cạnh (x, y) phải thuộc vào một cây bao trùm nhỏ nhất nào đó của G .
- 11.6.** Giả sử e là cạnh có trọng số lớn nhất trên một chu trình nào đó của đồ thị $G = (V, E)$. Chứng minh rằng, có cây bao trùm nhỏ nhất của đồ thị $G' = (V, E \setminus \{e\})$ mà cũng là cây bao trùm nhỏ nhất của G . Nghĩa là, có cây bao trùm nhỏ nhất của G không chứa cạnh e .

- 11.7.** Giáo sư P đưa ra thuật toán tìm cây bao trùm nhỏ nhất của đồ thị có trọng số như sau:

Chia tập đỉnh của đồ thị thành hai phần rời nhau, ta nhận được hai đồ thị con. Tìm cây bao trùm nhỏ nhất của hai đồ thị con này. Sau đó, chọn một cạnh có trọng số bé nhất nối hai cây này lại thì ta được cây bao trùm nhỏ nhất của đồ thị đã cho.

Thuật toán của Giáo sư P đúng hay sai? Nếu sai, hãy cho phản ví dụ.

- 11.8.** Đoàn Thanh niên ĐHQG HN tổ chức đêm lửa trại tại Hoà Lạc. Mỗi chi đoàn cắm một trại trên một khu đất rộng. Tất cả có n trại. Mỗi trại có tọa độ (x_i, y_i) , $i = 1, 2, \dots, n$. Để cấp điện thắp sáng cho các trại, Ban tổ chức phải mắc dây điện từ trạm biến áp có tọa độ (x_0, y_0) nối tiếp tới các trại. Hãy xây dựng giúp Ban tổ chức một sơ đồ mạng điện, sao cho số dây điện cần dùng là ít nhất.

- 11.9.** Chứng minh rằng, nếu cây nhị phân đầy đủ có l lá thì:

- Số đỉnh trong của cây là $l - 1$.
- Chiều cao của cây bằng $\log_2 l$.

- 11.10.** Xây dựng cây biểu thức cho các biểu thức sau đây:

a) $(A + B)/(C - D^2E)$.

b) $A(B - C/2) + (D + E)(F - G) - H$.

- 11.11.** Xây dựng bộ mã Huffman cho tập ký hiệu sau:

Ký hiệu	a	b	c	d	e	f	g
Tần suất (%)	21	26	9	8	10	22	4

TÀI LIỆU THAM KHẢO

1. A.V. Aho, J. E. Hopcroft & J.D. Ullman
The Design and Analysis of Computer Algorithms.
Addison – Wesley Publishing Co, 1974.
2. Claude Berge
Théorie des Graphes et Les Applications.
Dunod, Paris, 1967 (Bản dịch tiếng Việt của Nguyễn Hữu Nguyên và Nguyễn Văn Vy, NXB Khoa học và Kỹ thuật, 1971).
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein
Introduction to Algorithms.
MIT Press, 2001.
4. Nguyễn Cam và Chu Đức Khánh
Lý thuyết Đồ thị.
NXB Thành phố Hồ Chí Minh, 1999.
5. Vũ Đình Hoà
Một số kiến thức cơ sở về Graph hữu hạn.
NXB Giáo dục, 2002.
6. Nguyễn Hữu Ngự
Lý thuyết Đồ thị.
NXB Đại học Quốc gia Hà Nội, 2001.
7. Đặng Huy Ruận
Lý thuyết Đồ thị và ứng dụng.
NXB Khoa học và Kỹ thuật, 2000.
8. Hoàng Chí Thành
Giáo trình Tổ hợp.
NXB Đại học Quốc gia Hà Nội, 1999.
9. Nguyễn Tô Thành và Nguyễn Đức Nghĩa
Toán rời rạc.
NXB Giáo dục, 1997.

MỤC LỤC

Trang

LỜI NÓI ĐẦU	3
<i>Chương 1. ĐẠI CƯƠNG VỀ ĐỒ THỊ</i>	5
1.1. Khái niệm đồ thị	5
1.1.1. Định nghĩa đồ thị	5
1.1.2. Đường đi và chu trình	7
1.1.3. Đồ thị con và đồ thị riêng	8
1.1.4. Sự đẳng hình của các đồ thị	8
1.1.5. Các cách biểu diễn đồ thị trong máy tính	9
1.2. Một số tính chất về đường đi trên đồ thị	11
1.3. Bậc của đỉnh và tính liên thông của đồ thị	13
Bài tập	18
<i>Chương 2. HÀM GRUNDY TRÊN ĐỒ THỊ</i>	21
2.1. Hàm Grundy	21
2.2. Tổng của các đồ thị	24
Bài tập	27
<i>Chương 3. CÁC TẬP HỢP ĐẶC BIỆT TRÊN ĐỒ THỊ</i>	29
3.1. Tập ổn định trong	29
3.2. Tập ổn định ngoài	32
3.3. Nhân của đồ thị	34
3.4. Ứng dụng nhân vào lý thuyết trò chơi	40
Bài tập	43
<i>Chương 4. CHU SỐ VÀ SẮC SỐ CỦA ĐỒ THỊ</i>	45
4.1. Chu số của đồ thị	45
4.2. Sắc số của đồ thị	51
Bài tập	58
<i>Chương 5. CẶP GHÉP VÀ ĐỒ THỊ HAI PHẦN</i>	59
5.1. Tập đỉnh tựa và cặp ghép	59
5.2. Đồ thị hai phần	60
Bài tập	68

Chương 6. CÁC THUẬT TOÁN DUYỆT ĐỒ THỊ	69
6.1. Các thuật toán duyệt đồ thị	69
6.1.1. Thuật toán duyệt đồ thị	69
6.1.2. Duyệt đồ thị theo chiều sâu (Depth-First Search)	69
6.1.3. Duyệt đồ thị theo chiều rộng (Breadth-First Search)	72
6.2. Một số ứng dụng của thuật toán duyệt đồ thị	73
6.2.1. Bài toán tìm đường đi	73
6.2.2. Bài toán tìm các mảng liên thông	74
Bài tập	75
Chương 7. CHU TRÌNH EULER VÀ CHU TRÌNH HAMILTON	76
7.1. Chu trình Euler	76
7.2. Chu trình Hamilton	82
Bài tập	89
Chương 8. BÀI TOÁN ĐƯỜNG ĐI NGẮN NHẤT	91
8.1. Bài toán đường đi ngắn nhất	91
8.2. Bài toán đường đi có trọng số bé nhất	93
8.3. Thuật toán Dijkstra tìm đường đi ngắn nhất	93
8.4. Đường đi trên đồ thị phi chu trình	96
8.5. Đường đi ngắn nhất giữa tất cả các cặp đỉnh	97
8.6. Tâm của đồ thị	99
Bài tập	101
Chương 9. MẠNG VẬN TẢI	102
9.1. Bài toán luồng lớn nhất	102
9.1.1. Mạng vận tải	102
9.1.2. Luồng qua mạng	102
9.1.3. Bài toán luồng lớn nhất	104
9.1.4. Thuật toán Ford – Fulkerson tìm luồng lớn nhất	104
9.2. Một số ứng dụng của bài toán luồng lớn nhất	109
9.2.1. Bài toán luồng nhỏ nhất	109
9.2.2. Bài toán luồng trên mạng có nhiều đỉnh phát và đỉnh thu	111
9.2.3. Bài toán tìm cặp ghép lớn nhất của đồ thị hai phần	112
9.2.4. Bài toán vận tải với khả năng thông qua của các cạnh và các đỉnh	113
Bài tập	114

<i>Chương 10.</i>	ĐỒ THỊ PHẪNG	115
10.1.	Bài toán ba biệt thự và ba nhà máy	115
10.2.	Đồ thị phẳng	115
10.3.	Các điều kiện cho tính phẳng của đồ thị	117
10.4.	Sắc số của đồ thị phẳng	120
	Bài tập	122
<i>Chương 11.</i>	CÂY VÀ MỘT SỐ ỨNG DỤNG	123
11.1.	Cây	123
11.2.	Cây bao trùm của đồ thị	125
11.2.1.	Cây bao trùm	125
11.2.2.	Một số ứng dụng của cây bao trùm	129
11.3.	Cây bao trùm nhỏ nhất	131
11.3.1.	Bài toán cây bao trùm nhỏ nhất	131
11.3.2.	Các thuật toán tìm cây bao trùm nhỏ nhất	131
11.4.	Cây bao trùm lớn nhất	136
11.5.	Cây phân cấp	136
11.5.1.	Định nghĩa cây phân cấp	136
11.5.2.	Các cách duyệt cây	139
11.6.	Cây nhị phân	140
11.6.1.	Cây biểu thức	141
11.6.2.	Cây mã tiền tố	142
11.6.3.	Cây mã Huffman	143
	Bài tập	146
	TÀI LIỆU THAM KHẢO	148

Chịu trách nhiệm xuất bản :

Chủ tịch HĐQT kiêm Tổng Giám đốc NGÔ TRẦN ÁI
Phó Tổng Giám đốc kiêm Tổng biên tập NGUYỄN QUÝ THAO

Tổ chức bản thảo và chịu trách nhiệm nội dung :

Chủ tịch HĐQT kiêm Giám đốc Công ty CP Sách ĐH – DN
TRẦN NHẬT TÂN

Biên tập và sửa bản in :

ĐỖ HỮU PHÚ

Trình bày bìa :

HOÀNG MẠNH DŨA

Chế bản :

QUANG CHÍNH

ĐỒ THI VÀ CÁC THUẬT TOÁN

Mã số: 7B664M7 - DAI

In 1.000 bản, khổ 16 x 24 cm, tại Công ty In - Thương mại TTXVN
70/342 Khương Đình - Hạ Đình - Thanh Xuân - Hà Nội.

Số xuất bản: 11-2007/CXB/38-2117/GD.

In xong và nộp lưu chiểu tháng 02 năm 2007.



CÔNG TY CỔ PHẦN SÁCH ĐẠI HỌC DẠY NGHỀ
HEVOBCO

25 HÀN THUYỀN – HÀ NỘI

Website : www.hevobco.com.vn

**TÌM ĐỌC SÁCH THAM KHẢO VỀ TOÁN – TIN
CỦA NHÀ XUẤT BẢN GIÁO DỤC**

1. Đồ thị và các thuật toán	PGS. TS. Hoàng Chí Thành
2. Hướng dẫn giải Bài tập Toán rời rạc	PGS. TS. Đỗ Đức Giáo
3. Phương pháp tính	Tạ Văn Đĩnh
4. Toán học tính toán	Doãn Tam Hoà
5. Đại số tuyến tính và hình học giải tích	PGS. TS. Trần Trọng Huệ
6. Cấu trúc máy tính	TS. Trần Quang Vinh
7. Lập trình bằng ngôn ngữ Assembly cho máy tính	TS. Nguyễn Mạnh Giang

Bạn đọc có thể mua tại các Công ty Sách – Thiết bị trường học ở địa phương hoặc các Cửa hàng của Nhà xuất bản Giáo dục :

Tại Hà Nội : 25 Hàn Thuyên ; 187B Giảng Võ ; 23 Tràng Tiền ; 232 Tây Sơn.

Tại Đà Nẵng : 15 Nguyễn Chí Thanh ; 62 Nguyễn Chí Thanh.

Tại Thành phố Hồ Chí Minh : 240 Trần Bình Trọng – Quận 5 ;

104 Mai Thị Lựu – Quận 1 ;

Cửa hàng 451B, 453 Hai Bà Trưng – Quận 3.

Tại Thành phố Cần Thơ : Số 5/5, đường 30/4



Giá: 18.000 đ