ĐẠI SỐ
LÝ THUYẾT ĐỒ THỊ

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

CHƯƠNG 1.

CÁC KHÁI NIỆM CƠ BẢN VỀ ĐỒ THỊ.

1.1 ĐỊNH NGHĨA & THÍ DỤ.

1.1.1 ĐỊNH NGHĨA.

1.1.1.1 Đồ thị có định hướng.

Một đồ thị $G = G(X,U)$ được xác định bởi

- Tập hữu hạn $X = \{x_1, x_2, ..., x_n\}$ tập các đỉnh hay nút.
- Tập $U = \{u_1, u_2, ..., u_n\} \subset X \times X$ tập các cung (cạnh).

Đối với một cung $u = (x_i, x_j)$, x_i là đỉnh đi, x_j là đỉnh đến (hay còn gọi là gốc và đích). Cung u đi từ x_i và đến x_j.

Cung u được biểu diễn một cách hình học như sau :

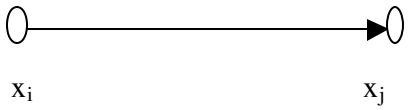


FIG.1.1. Cung $u=(x_i, x_j)$

Một cung (x_i, x_i) được gọi là một **vòng** (**khuyên**).

Một **p-đồ thị** là một đồ thị trong đó không có quá p cung dưới dạng (i,j) giữa hai đỉnh bất kỳ.

Thí dụ.

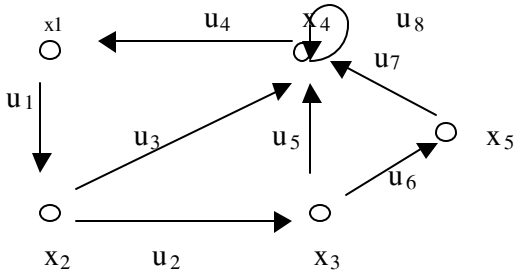


FIG. 1.2. Đồ thị xác định bởi (X,U),

$X = \{x_1, x_2, x_3, x_4, x_5\}$; $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.1.1.2 Đồ thị không định hướng.

Khi khảo sát một vài tính chất, sự định hướng của các cung không đóng một vai trò gì. Ta chỉ quan tâm đến sự hiện diện của các cung giữa hai đỉnh mà thôi (không cần định rõ thứ tự). Một cung không định hướng được gọi là **cạnh.** Đối với một cạnh $u = (x_i,x_j)$, u được gọi là CẠNH TỚI của hai đỉnh x_i và x_j.

Thí dụ.

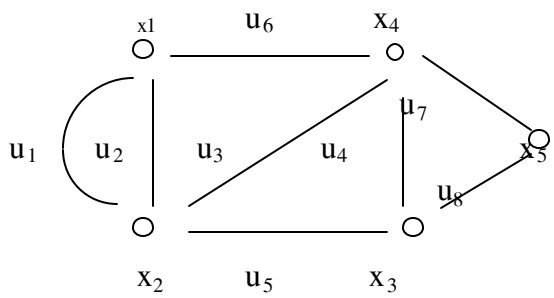


FIG. 1.3. Đồ thị xác định bởi (X,U),
$X = \{x_1, x_2, x_3, x_4, x_5\}$; $U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$

Một đồ thị được gọi là **đa đồ thị** nếu có nhiều hơn một cạnh giữa hai đỉnh.

Một đồ thị được gọi là **đơn** nếu:

1. Không phải là đa đồ thị ;
2. Không tồn tại một vòng nào.

Hai cạnh u và v được gọi là **song song** khi chúng cùng là cạnh tới của hai đỉnh phân biệt. Ký hiệu $u \mid v$.

Theo thí dụ trên, ta có $u_1 \mid u_2$

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.1.1.3 Một số định nghĩa cơ bản.

- **ÁNH XẠ ĐA TRỊ.**
  - x_j được gọi là ĐỈNH SAU (SUCCESSEUR) của x_i nếu $(x_i,x_j) \in U$; Tập các đỉnh sau của x_i ký hiệu là $\Gamma(x_i)$.
  - x_j được gọi là ĐỈNH TRƯỚC (PREDECESSEUR) của x_i nếu $(x_j,x_i) \in U$; Tập các đỉnh trước của x_i ký hiệu là $\Gamma^{-1}(x_i)$.
  - Ánh xạ Γ được định nghĩa :với mọi phần tử của X, tương ứng với một tập con của X được gọi là một ÁNH XẠ ĐA TRỊ.
  - Đối với một 1-đồ thị, G có thể hoàn toàn xác định bởi (X,Γ), đây là một ký hiệu cơ sở thường dùng trong cấu trúc dữ liệu : DANH SÁCH KỀ.

THÍ DỤ. Trong đồ thị được định nghĩa ở hình vẽ sau. $X = \{x_1,x_2,x_3,x_4,x_5\}$; $\Gamma(x_1) = x_2$; $\Gamma(x_2) = \{x_3,x_4\}$; $\Gamma(x_3)=\{x_4,x_5\}$; $\Gamma(x_4)=\{x_1\}$; $\Gamma(x_5)=\{x_4\}$.

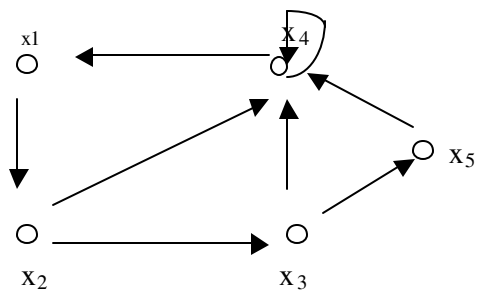


FIG. 1.4. Đồ thị xác định bởi (X,Γ)

- **KỀ.**
  - Hai đỉnh được gọi là kề nếu chúng được nối bởi một cung (cạnh).
  - Hai cung (cạnh) được gọi là kề nếu chúng có ít nhất một đỉnh chung.

- **BẬC CỦA ĐỈNH.**
  - **Nửa bậc ngoài** của đỉnh x_i, ký hiệu $d^+(x_i)$ là số các cung khởi đầu từ (hay đi ra từ) x_i . Ta có $d^+(x_i) = \text{card}(\Gamma(x_i))$. (ký hiệu card(A) chỉ số phần tử của tập A).

  - **Nửa bậc trong** của đỉnh x_i, ký hiệu $d^-(x_i)$ là số các cung kết thúc tại (hay đi vào từ) x_i . Ta có $d^-(x_i)=\text{card}(\Gamma^{-1}(x_i))$.

  - **Bậc của đỉnh** x_i, $d(x_i) = d^+(x_i) + d^-(x_i)$. Bậc của một đỉnh trong một đồ thị không định hướng là tổng số các cạnh tới của nó.

Bậc của một đỉnh có vòng được cộng thêm 2 cho mỗi vòng.

THÍ DỤ. [xem FIG. 1.4].

$d^+(x_2)= 2$; $d^-(x_2)= 1$; $d(x_2)=3$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

$d^+(x_4)= 1$; $d^-(x_4)= 3$; $d(x_4)=6$. (Vì tại đỉnh x_4 có một vòng).

- Đỉnh có bậc = 0 được gọi là **đỉnh cô lập.**

- Đỉnh có bậc = 1 được gọi là **đỉnh treo** và cung (cạnh) tới của nó được gọi là **cạnh treo**.

- **ĐỊNH LÝ** (công thức liên hệ giữa bậc và số cạnh).

  1. Tổng bậc các đỉnh = 2 x số cạnh.

  2. Xét đồ thị có định hướng G = (X, U). Ta có

     $\sum d^+(x) = \sum d^-(x) =$ card(U) (số cung).

  CHỨNG MINH. Truy chứng theo đỉnh.

- **HỆ QUẢ**. Số đỉnh bậc lẻ là số chẵn.

  CHỨNG MINH.

  $\sum$ d(đỉnh bậc lẻ) + $\sum$ d(đỉnh bậc chẵn) = 2 x số cạnh.

- **ĐỒ THỊ BÙ.**

  G = (X, U) và $\overline{G} = (X, \overline{U})$. $(x_i,x_j) \in U \Rightarrow (x_i,x_j) \notin \overline{U}$ et $(x_i,x_j) \notin U \Rightarrow (x_i,x_j) \in \overline{U}$.
  $\overline{G}$ được gọi là đồ thị bù của G.

- **ĐỒ THỊ RIÊNG PHẦN (BỘ PHẬN).**
  G=(X,U) và $U_p \subset U$. $G_p=(X,U_p)$ là một đồ thị riêng phần của G ;

- **ĐỒ THỊ CON.**
  G=(X,U) và $X_s \subset X$. $G_s=(X_s,V)$ là một đồ thị con của G; trong đó V là thu hẹp của hàm đặc trưng của U trên X_s.

  $V=\{(x,y)/(x,y) \in U \cap X_s \times X_s\}$. $\forall x_i \in X_s$, $\Gamma_s(x_i)=\Gamma(x_i)\cap X_s$.

- **ĐỒ THỊ CON RIÊNG PHẦN.** Tổng hợp hai định nghĩa trên.

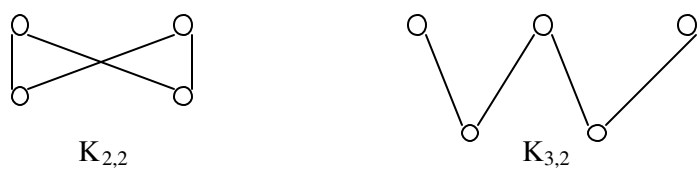
THÍ DỤ. Mạng giao thông đường bộ cả nước.
- Mạng xe bus : đồ thị riêng phần.
- Mạng giao thông đường bộ T.P. Hồ Chí Minh: đồ thị con.
- Mạng xe bus T.P. Hồ Chí Minh: đồ thị con riêng phần.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

- **ĐỒ THỊ** đối xứng : $(x_i,x_j) \in U \Rightarrow (x_i,x_i) \in U$.
- **ĐỒ THỊ** phản đối xứng : $(x_i,x_j) \in U \Rightarrow (x_j,x_i) \notin U$.
- **ĐỒ THỊ** phản chiếu : $(x_i,x_i) \in U, \forall x_i \in U$.
- **ĐỒ THỊ** bắc cầu : $(x_i,x_j) \in U, (x_j,x_k) \in U \Rightarrow (x_i,x_k) \in U$.
- **ĐỒ THỊ** đầy đủ : $(x_i,x_j) \notin U \Rightarrow (x_j,x_i) \in U$ (có duy nhất một cạnh giữa hai đỉnh). Một đồ thị đủ có n đỉnh sẽ có n(n-1)/2 cạnh. Ký hiệu K_n.
- **CLIQUE** :Tập các đỉnh của một đồ thị con đầy đủ.
- **ĐỒ THỊ HAI PHẦN (LƯỠNG PHÂN)** G=(X,U) nếu :
  1. X phân hoạch thành X_1 và X_2.
  2. $\forall (x_1,x_2) \in U$ thì $x_1 \in X_1$, $x_2 \in X_2$.

  Nếu $Card(X_1) = n$, $Card(X_2) = m$, ký hiệu $K_{n,m}$.

  Thí dụ : Đồ thị sau lưỡng phân, nhưng không đầy đủ.



- **ĐỀU.** Là đồ thị mà mọi đỉnh có cùng bậc.

THÍ DỤ.

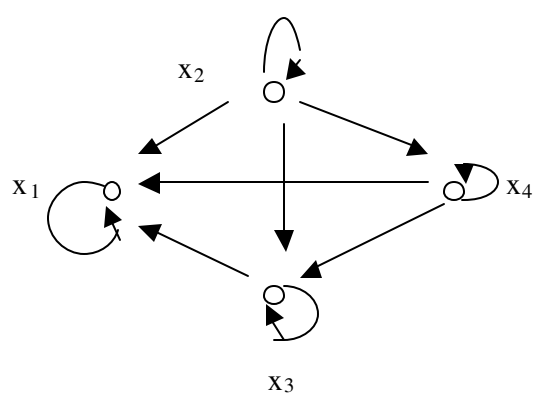


FIG. 1.5. Đồ thị phản chiếu , phản đối xứng, bắc cầu và đầy đủ.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.1.2 THÍ DỤ.

- **THÍ DỤ 1**. Đường đi ngắn nhất.

Bài toán 1. Cho một đồ thị có định hướng, $G = (X,U)$, một định giá $v : U \to R$ và s, t là hai đỉnh phân biệt của X.

Bài toán đặt ra. Tìm đường đi ngắn nhất giữa s và t ?

Lời giải. Thuật giải Dijkstra, Bellman-Ford (xem Chương 3).

`

- **THÍ DỤ 2**. Cây phủ tối thiểu.

Xét bài toán trên một mạng, chẳng hạn mạng cung cấp điện, nước từ một nguồn duy nhất.

Bài toán 2. Một đồ thị không định hướng $G = (X,U)$, một hàm định giá trọng lượng $v : U \to R^+$ và hai đỉnh phân biệt s, t của X.

Bài toán đặt ra. Tìm một cây phủ với trong lượng tối thiểu ?

Lời giải : Thuật giải Kruskal, Prim (xem Chương 2).

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.2 BIỂU DIỄN ĐỒ THỊ.

Có rất nhiều cách để biểu diễn đồ thị. Tuy nhiên, các cách biểu diễn này không tương đương với nhau theo quan điểm của các thuật toán. Người ta, phân biệt một vài cách biểu diễn chính, chẳng hạn biểu diễn bằng ma trận kề, ma trận tới đỉnh – cung (hay đỉnh – cạnh trong trường hợp không định hướng) và bằng danh sách kề.

1.2.1 Biểu diễn bằng cách sử dụng các Bảng.

1.2.1.1. Ma trận kề.

Xét một 1 - đồ thị có n đỉnh. Ma trận kề là một ma trận (n x n) có n hàng tương ứng với các đỉnh khởi đầu và n cột tương ứng với các đỉnh kết thúc, được định nghĩa như sau :

x_{ij} = 1 (True) nếu có một cung (cạnh) nối x_i và x_j.
= 0 (False) ngược lại.

THÍ DỤ.

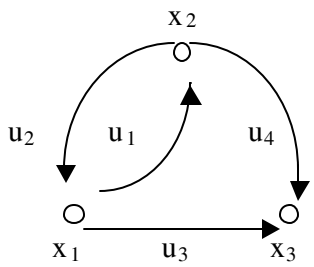


FIG.1.6. 1. Đồ thị.

Ma trận kề của đồ thị này như sau :

	x_1	x_2	x_3 ← kết thúc
x_1	0	1	1
x_2	1	0	1
x_3	0	0	0

↑
khởi đầu

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.2.1.2. Ma trận tới đỉnh – cung (đỉnh – cạnh).

- Dòng ↔ đỉnh.
- Cột ↔ cung (cạnh).

Cho đồ thị G = (X, U). Một ma trận tới A = [a_{ij}] được định nghĩa như sau :
Nếu cạnh u = (x_i, x_j) Î U thì trên cột u, a_{iu} = 1, a_{ju} = -1, ngược lại thì có giá trị 0.

THÍ DỤ. Đối với 1. Đồ thị ở hình FIG .1.6. ta có :

	U_1	u_2	u_3	u_4
x1	1	-1	1	0
x_2	-1	1	0	1
x_3	0	0	-1	-1

CHÚ Ý : Tổng các dòng bằng không (một cung có đỉnh gốc và một đỉnh kết thúc).
Tất cả các ma trận con vuông đều có định thức bằng 1, -1 hay 0.

Có một cách khác cho ma trận tới như sau :
Cho đồ thị G = (X, U). Một ma trận tới A = [a_{ij}] được định nghĩa như sau :

a_{iu} = 1 nếu u = (x_i,x_j) ∈ U
= 0 ngược lại.

THÍ DỤ. Đối với 1. Đồ thị ở hình FIG .1.6. ta có :

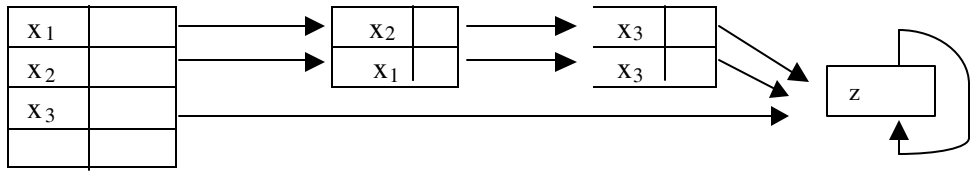
	u_1	u_2	u_3	u_4
x1	1	0	1	0
x_2	0	1	0	1
x_3	0	0	0	0

CHÚ Ý : Tổng các dòng bằng số các cung tới.

1.2.2 Biểu diễn bằng cách sử dụng các con trỏ.

Lợi ích của cách biểu diễn bằng con trỏ hay Danh sách kề (nhờ vào ánh xạ đa trị Γ) là giảm thiểu chổ trong bộ nhớ.

THÍ DỤ. Đối với 1.đồ thị của hình FIG.1.6. ta có :



Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.3 PHÉP DUYỆT ĐỒ THỊ. (Parcours de graphes).

Nhiều bài toán trên đồ thị cần khảo sát sự vét kiệt các đỉnh và các cung (cạnh) của đồ thị. Có 2 cách duyệt đồ thị : **phép duyệt theo chiều sâu** (Parcours en profondeur) và **phép duyệt theo chiều rộng** (Parcours en largeur).

1.3.1. DUYỆT THEO CHIỀU SÂU.

NGUYÊN LÝ :
Khởi từ một đỉnh, đi theo các cung (cạnh) xa nhất có thể. Trở lại đỉnh sau của cạnh xa nhất, tiếp tục duyệt như trước, cho đến đỉnh cuối cùng.

Thí dụ. Ta có đồ thị theo hình vẽ sau :

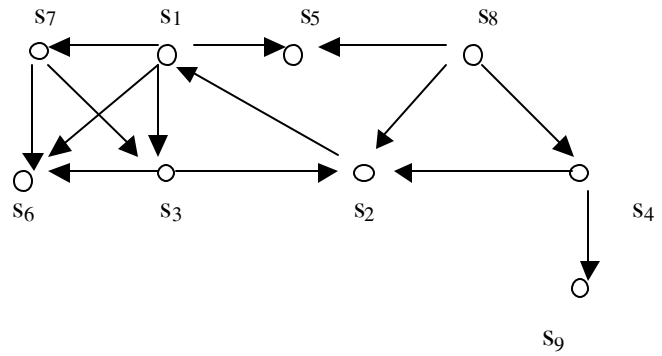


FIG. 1.7.

Phép duyệt theo chiều sâu thực hiện trên đồ thị ở hình FIG.1.7 như sau :

- Khởi từ đỉnh s_1. Đỉnh đầu tiên được duyệt là s_3.
- Khởi từ đỉnh s_3. Đỉnh được duyệt là s_2. Đỉnh sau của s_3 là s_6.
- Khởi từ đỉnh s_6. Đỉnh sau của s_1 là s_5.
- Khởi từ đỉnh s_5. Đỉnh sau của s_1 là s_7.
- Khởi từ đỉnh s_7.
- Khởi từ đỉnh s_4. Đỉnh được duyệt là s_9.
- Khởi từ đỉnh s_8.
- Kết thúc vì tất cả các đỉnh đã được duyệt.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Ký hiệu :

s[k], k : 1..n là tập đỉnh có n phần tử, được đánh số thứ tự từ 1 đến n.
Mark[k], k : 1..n là hàm nguyên :
= 1 nếu đỉnh đã được duyệt (có nghĩa đã được đánh dấu),
= 0 ngược lại.
Ma trận kề a, được định nghĩa như sau :
a[i,j] = 1, nếu (i,j) là một cung (cạnh) của đồ thị G.
= 0 ngược lại.

Dạng đệ qui

Chương trình chính :

```
For (int i =1; i ≤ n ;i++) Mark[i] = 0 ;
For (int i =1; i ≤ n ;i++)  if( Mark[i] == 0) then DFS(i) ;
```

Thủ tục đệ qui : Duyệt theo chiều sâu bắt đầu từ đỉnh k.

```
Thủ tục DFS(int k) ;
{
    Mark[k] = 1
    // Duyệt các đỉnh trong ma trận kề của đỉnh k
    For (int j =1; j ≤ n ;j++)
      if (Mark[j] == 0 && a[k][j]==1) DFS(j) ;
 } End DFS
```

Độ phức tạp của giải thuật : Đồ thị có n đỉnh và m cung(cạnh).

- Trường hợp lưu trữ đồ thị dưới dạng ma trận kề : $O(n^2)$.
- Trường hợp lưu trữ đồ thị dưới dạng danh sách kề : O(max(n,p)).

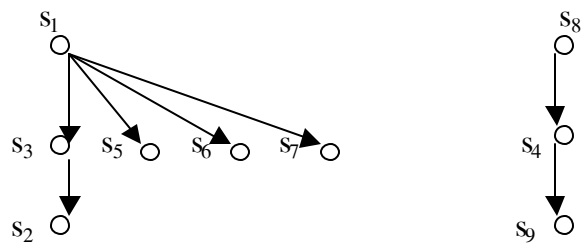
Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.6.2. DUYỆT THEO CHIỀU RỘNG.

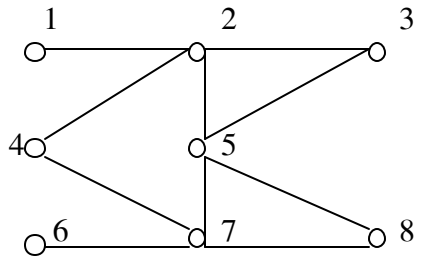
NGUYÊN LÝ :

- Khởi từ một đỉnh s bất kỳ, ta duyệt tất cả những đỉnh sau của S,tập $\Gamma^+(s)$ trong trường hợp đồ thị có định hướng (tập $\Gamma(s)$:tập tất cả các đỉnh kề của s trong trường hợp đồ thị không định hướng) .
- Sau đó xét $v \in \Gamma^+(s)$ (hay $\Gamma(s)$) và áp dụng lại cách duyệt giống như s.

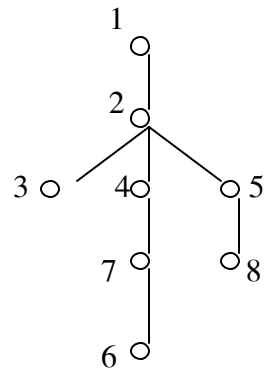
Thí dụ 1. Ta có đồ thị theo hình vẽ FIG 1.7. Duyệt theo chiều rộng như sau :



Thí dụ 2. Ta có đồ thị theo hình vẽ sau :



Duyệt theo chiều rộng như sau :



Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.4 TÍNH LIÊN THÔNG CỦA ĐỒ THỊ.

1.4.1. Dây chuyền - Chu trình.

Một dây chuyền trong một đồ thị không có định hướng là một dãy liên tiếp các cạnh, sao cho mỗi một cạnh có một đỉnh chung với cạnh tiếp theo. Một chu trình là một dây chuyền mà có ít nhất một cạnh có đỉnh khởi đầu và đỉnh kết thúc trùng nhau.

Thí dụ.

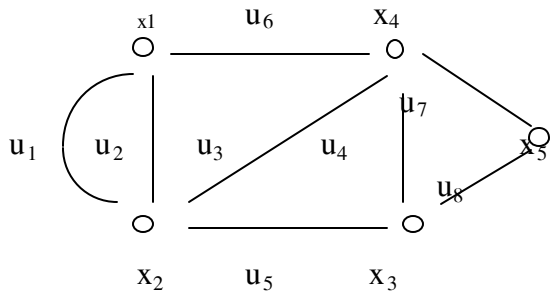


FIG.1.8. <u_5,u_2,u_6,u_7> là một dây chuyền, <u_4,u_7,u_8> là một chu trình.

1.4.2. Đường – Mạch.

Đường và mạch là khái niệm dây chuyền và chu trình trong trường hợp đồ thị có định hướng.

THÍ DỤ.

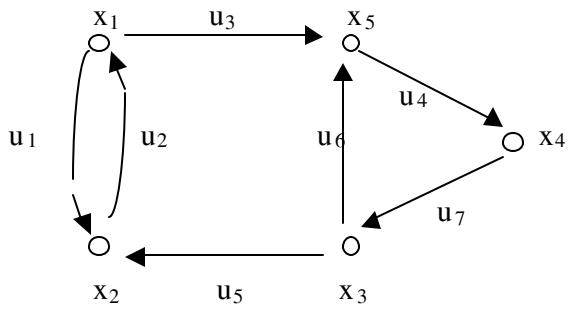


FIG.1.9. <u_5,u_2,u_3,u_4> là một đường, <u_4,u_7,u_6> là một mạch.

Tập con các đỉnh liên kết của một đường được gọi là BAO CHUYỀN.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Thuật ngữ **HÀNH TRÌNH** (PARCOURS) để chỉ nhóm lại các đường, các dây chuyền, các mạch và các chu trình. Một hành trình được gọi là :

- **SƠ CẤP** : Nếu Tất cả các đỉnh hợp thành đều phân biệt.
- **ĐƠN** : Nếu tất cả các cạnh đều phân biệt.
- **HAMILTON** : Đi qua đúng một lần đối với mỗi đỉnh của đồ thị.
- **EULER** : Đi qua đúng một lần tại mỗi cạnh của đồ thị.
- **TIỀN HAMILTON**: Đi qua it nhất một lần đối với mỗi đỉnh của đồ thị.
- **TIỀN EULER (CHINOIS)** : Đi qua ít nhất một lần tại mỗi cạnh của đồ thị.

1.4.3. Tính liên thông .

Một đồ thị không định hướng được gọi là LIÊN THÔNG (CONNEXE) nếu với mọi cặp đỉnh đều có đường nối.

THÀNH PHẦN LIÊN THÔNG là một đồ thị con liên thông tối đại.

THÍ DỤ :

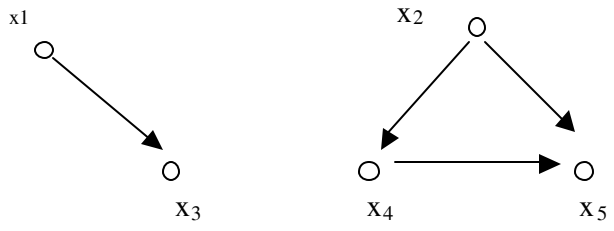


FIG.1.10. Đồ thị có hai thành phần liên thông.

ĐỊNH LÝ 1

Một đồ thị là liên thông nếu và chỉ nếu nó có một thành phần liên thông.

Chứng minh. Hiễn nhiên.

ĐỊNH LÝ 2.
Một đồ thị có đúng hai đỉnh bậc lẻ thì phải có một đường nối hai đỉnh này.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Chứng minh. Chứng minh bằng phản chứng.

1.4.4. Liên thông mạnh.

Một đồ thị có định hướng được gọi là liên thông mạnh nếu với mọi cặp đỉnh phân biệt có một đường nối chúng.

Một thành phần liên thông mạnh (CFC) là đồ thị con tối đại liên thông mạnh.

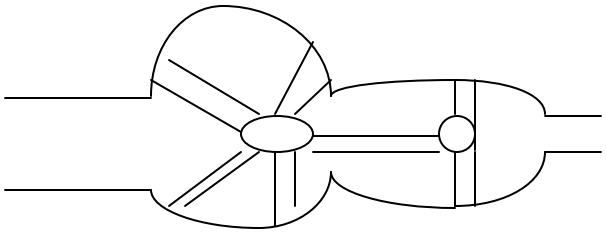
ĐỊNH LÝ

Một đồ thị là liên thông nếu và chỉ nếu nó có một thành phần liên thông mạnh.

Chứng minh. Hiển nhiên.

1.5 ĐỒ THỊ EULER.

1.5.1. Bài toán 7 chiếc cầu.



Đây là tình huống có thật ở Konigsberg (nước Đức), có hai vùng bị ngăn cách bởi một dòng sông và có hai cù lao ở giữa sông, 7 chiếc cầu nối những vùng này với nhau như minh họa trong hình vẽ trên. Người dân trong vùng thách đố nhau là thử tìm cách xuất phát từ một vùng đi dạo qua mỗi chiếc cầu đúng một lần và trở về nơi xuất phát. Năm 1736, nhà toán học Euler đã mô hình hóa bài toán nàybằng một đồ thị vô hướng với mỗi đỉnh ứng với một vùng, mỗi cạnh ứng với một chiếc cầu. Bài tóan được phát biểu lại cho đồ thị trong hình vẽ bên dưới, hãy tìm một đường đi trong đồ thị đi qua một lần trong tất cả các cạnh và sau đó trở về đỉnh xuất phát. Việc giải bài toán đưa đến các định lý EULER.

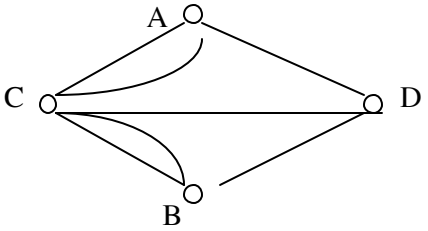


FIG. 1.11. Bài toán 7 chiếc cầu.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.5.2. Định nghĩa.

Đồ thị không định hướng (có định hướng) EULER là đồ thị không định hướng (có định hướng) có chứa một mạch (chu trình) EULER.

Thí dụ.

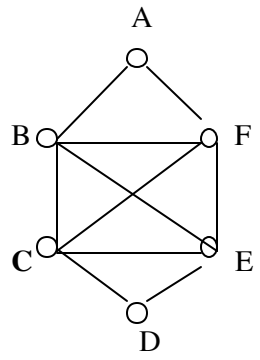


FIG. 1.12. <ABEDCEFCBFA.> là mạch EULER.

Đồ thị sau đây không có mạch EULER, nhưng có các đường EULER.

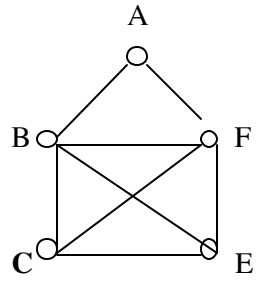


FIG. 1.13. <EBACBDCED> là một đường EULER.

1.5.3. Định lý EULER.

- **Định lý 1.** Một đồ thị không định hướng, liên thông là đồ thị EULER nếu và chỉ nếu mọi đỉnh của G có bậc chẵn.

- **Định lý 2.** Cho G= (X,U) là một đồ thị có định hướng, liên thông mạnh. Khi đó G là đồ thị Euler nếu và chỉ nếu ta có :

  $d^+(x) = d^-(x)$ với mọi đỉnh x.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

- **Định lý 3.** Cho G=(X,U) là một đồ thị không định hướng, liên thông. Khi đó G có đường Euler nếu và chỉ nếu G có đúng 2 đỉnh có bậc lẻ.

Thí dụ.

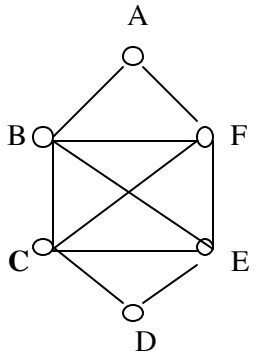


FIG.1.14. Đồ thị không định hướng có mọi đỉnh có bậc chẵn nên là đồ thị EULER

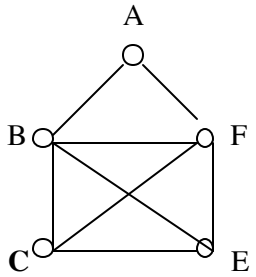


FIG. 1.15. Đồ thị có 2 đỉnh bậc lẻ nên không phải là đồ thị Euler, thỏa định lý 3 nên đồ thị sẽ có một đường Euler.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1.6 ĐỒ THỊ HAMILTON.

Khái niệm đường đi Hamilton được xuất phát từ bài toán « Xuất phát từ một đỉnh của khối thập nhị diện đều, hãy đi dọc theo các cạnh của khối đó sao cho đi qua đúng một lần tất cả các đỉnh của đồ thị ». Bài toán này được nhà Toán học Hamilton đưa vào năm 1859.

1.6.1. Định nghĩa.

Đồ thị HAMILTON là đồ thị có chứa một chu trình HAMILTON.

1.6.2. Tính chất.

- **Định lý 1.** Đồ thị đầy đủ là đồ thị Hamilton. Với n lẻ ≥ 3 thì K_n có $(n-1)/2$ chu trình Hamilton đôi một không có cạnh chung.

  Chứng minh. Hiễn nhiên.

- **Định lý 2.** Giả sử G là đồ thị đơn không có định hướng có n đỉnh, với $n \geq 3$. Nếu với mọi cặp đỉnh x, z sao cho z không là đỉnh kề của x , ta có :

  $$d(x) + d(z) \geq n$$

  Thì G là một đồ thị Hamilton.

  Chứng minh. Bài tập.

- **Định lý 3.** Giả sử G là đồ thị đơn không có định hướng có n đỉnh, với $n \geq 3$. Nếu với mọi đỉnh có bậc $\geq n/2$ thì G là một đồ thị Hamilton.

  Chứng minh. Suy từ định lý 2.

- **Định lý 4.** Gi ả sử G là đồ thị đơn không có định hướng có n đỉnh và m cạnh. Nếu $m \geq (n^2 - 3n + 6)/2$ thì G là một đồ thị Hamilton.

  Chứng minh. Áp dụng định lý 2.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

CHƯƠNG 2.

CẤU TRÚC CÂY.

2.1 ĐỊNH NGHĨA & THÍ DỤ.

2.1.1 CÂY.

Cây là một đồ thị không định hướng, liên thông và không có chu trình.

THÍ DỤ.

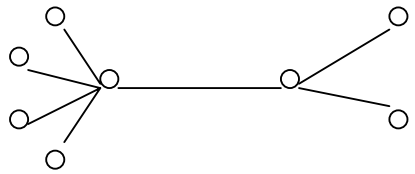


FIG. 2.1. Cây.

- Chiều dài của đường nối hai đỉnh lại với nhau được gọi là khoảng cách giữa hai đỉnh.

TÍNH CHẤT.

- Giữa hai đỉnh bất kỳ của một cây sẽ có duy nhất một dây chuyền nối chúng lại với nhau.
- Một cây n đỉnh sẽ có n –1 cạnh. Cộng thêm vào cây một cạnh giữa hai đỉnh bất kỳ sẽ tạo nên một chu trình duy nhất.

2.1.2 RỪNG.

Là một đồ thị không định hướng và không có chu trình (Không liên thông mạnh) Mỗi thành phần liên thông của một rừng là một cây.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.1.3 CẤU TRÚC CÂY (CÂY CÓ GỐC).

Là một đồ thị có định hướng sao cho mỗi đỉnh đều có một đỉnh trước trừ một phần tử duy nhất không có , được gọi là **GỐC.** Với mọi đỉnh x thì có duy nhất một đường từ gốc đi đến x.

Xét một đỉnh x của một cây T có gốc là r :

- Một đỉnh y bất kỳ nằm trên đường hướng từ gốc đến x, được gọi là các **ĐỈNH TRƯỚC** (ANCETRE) của x, và x được là **ĐỈNH SAU** (DESCENDANT) của y.
- Nếu (x,y) là một cạnh của T, ta gọi x là **CHA** của y và y là **CON** của x. Hai đỉnh cùng cha được gọi là **ANH EM**. Một đỉnh không có con được gọi là LÁ. Những đỉnh không là LÁ được gọi là ĐỈNH TRONG.

- Chiều dài của đường từ gốc đến đỉnh được gọi là **độ sâu** của đỉnh đó.

- **Mức (Niveau)** của một đỉnh trong T là khoảng cách từ gốc đến x.
  - ❖ Mức của nút gốc = 0.
  - ❖ Mức của nút khác gốc = Mức của cây con nhỏ nhất chứa nó + 1.

- **Chiều cao hay độ sâu** (Hauteur, profondeur) của cây là giá trị lớn nhất của mức của các đỉnh trong cây.

- Nếu mỗi đỉnh trong cây có tối đa hai con, thì ta gọi đó là **cây nhị phân**.

- **Bậc của nút & bậc của cây (Degrée).**
  - ❖ Bậc của nút là số cây con của nút đó.
  - ❖ Bậc của cây là bậc lớn nhất của các nút của cây. Nếu cây có bậc là n, ta gọi là cây n-cành.

THÍ DỤ . Cây 3 – cành có gốc,với 8 đỉnh và có độ cao là 4.

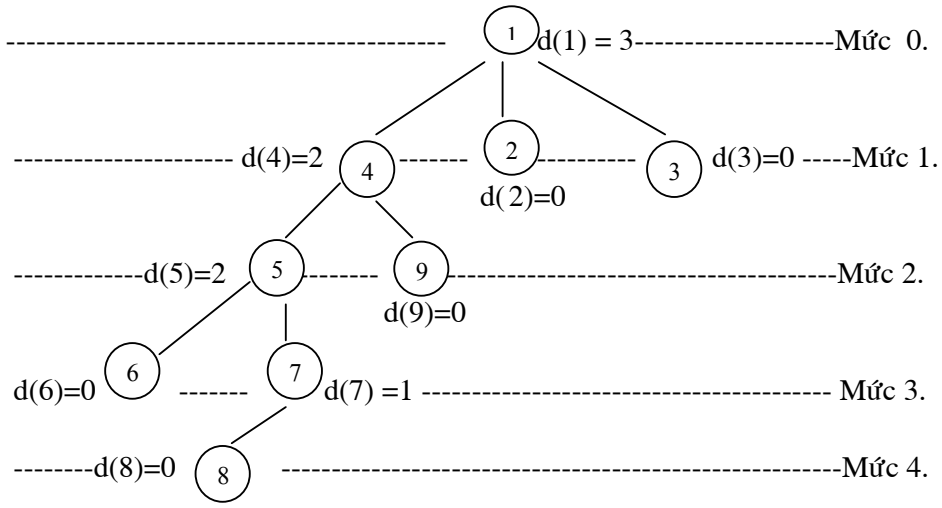


FIG.2.2. Cây có gốc.

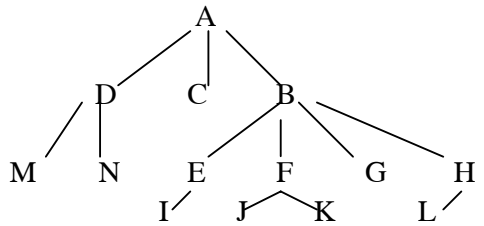
2.1.4. THÍ DỤ.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

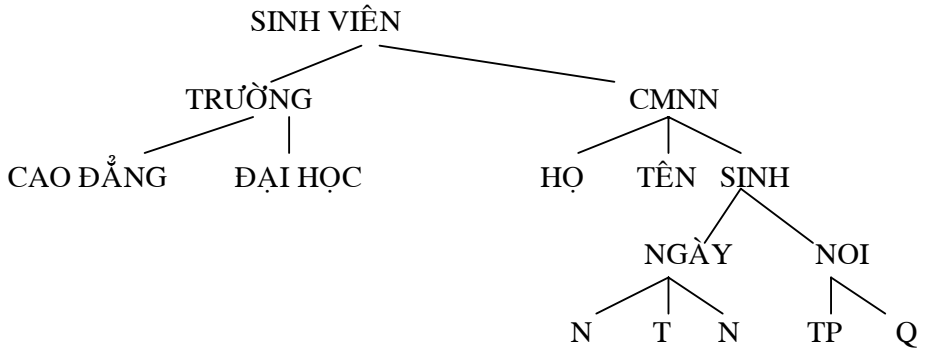
- Đôi khi ta có thể biểu diễn một quan hệ bao hàm thức của nhiều tập hợp bằng một cấu trúc cây.

  Thí dụ. Bao hàm của các tập hợp sau có thể biểu diễn thành cấu trúc cây như sau :

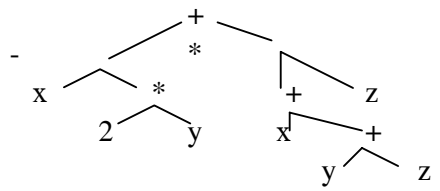
  B, C, D ⊂ A.
  E, F, G, H ⊂ B.
  M, N ⊂ D.
  I ⊂ E.
  J,K ⊂ F.
  L ⊂ H.



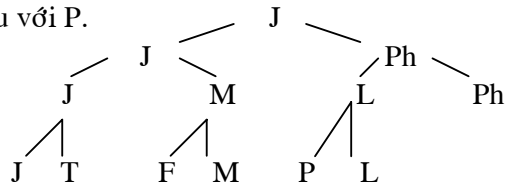
- **Một Biến có cấu trúc** có thể biểu diễn dưới dạng cây.



- **Biểu thức số học**. Biểu thức
  X = (x – (2* y) +((x+(y+z)) *z)
  có thể biểu diễn thành hình cây như sau :

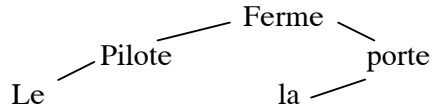


- **Vòng loại** trong một cuộc thi đấu bóng bàn.
  - ❖ Vòng 1. J đấu với T, F đấu với M, L đấu với P.
  - ❖ Vòng 2. J đấu với M, L đấu với Ph
  - ❖ Vòng 3. J đấu Ph.
  - ❖ Cuối cùng J thắng.

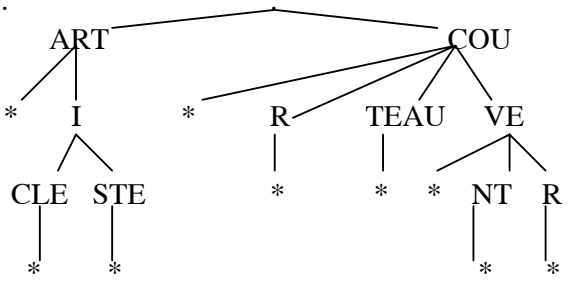


- **Câu trong ngôn ngữ tự nhiên** (hay trong ngôn ngữ lập trình)

  Đối với câu « Le Pilote ferme la porte »
  Có thể biểu diễn dưới dạng



- **Tự điển có thể tổ chức theo hình cây**.
  Chẳng hạn tự điển gồm các từ ART, ARTICLE, ASTISTE, COU, COUR, COUTEAU, COUVE, COUVENT, COUVER có thể biểu diễn theo hình vẽ sau. Ký tự «*» chỉ chấm dứt một từ. Chú ý, thứ tự ALPHABET theo thứ tự từ phải sang trái.



Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.2 TÍNH CHẤT CƠ BẢN.

2.2.1 ĐỊNH LÝ 1.

Cho G là một cây bậc n > 1. Các tính chất sau đây tương đương với nhau :

1. G liên thông và không có chu trình.
2. G liên thông và có n –1 cạnh.
3. G không có chu trình và có n – 1 cạnh.
4. G không có chu trình và nếu thêm vào một cạnh giữa hai đỉnh không kề sẽ tạo một chu trình duy nhất giữa chúng.
5. G liên thông tối thiểu(có nghĩa là nếu xóa đi một cạnh bất kỳ thì G không còn liên thông nữa)
6. Mọi cặp đỉnh có duy nhất dây chuyền nối chúng.

CHỨNG MINH. Bài tập.

2.2.2 ĐỊNH LÝ 2.

Một đồ thị G = (X,U) là một đồ thị có chứa một đồ thị riêng phần nếu và chỉ nếu G liên thông.

CHỨNG MINH. Bài tập.

2.2.3 ĐỊNH LÝ 3.

Mọi Cấu trúc cây là một cây.

CHỨNG MINH. Bài tập.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.3 CÂY NHỊ PHÂN.

2.3.1. ĐỊNH NGHĨA (THEO ĐỆ QUI).

Một cây nhị phân B hoăc là ∅ hoặc có dạng :

B = < O, B_1, B_2 > trong đó :

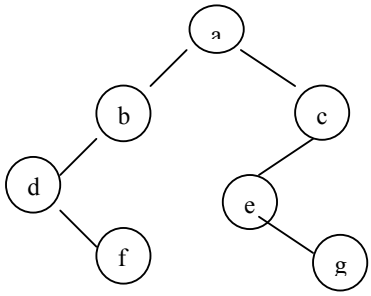
O : gốc,

B_1 : cây con trái và

B_2 : cây con phải.

2.3.2. BIỂU DIỄN CÂY NHỊ PHÂN.

THÍ DỤ.



❖ **SỬ DỤNG BẢNG**. Có thể định nghĩa kiểu dữ liệu như sau :

```
Type  Arbtab  = Array [1..n]  of  Record      v : t ;
                                              G : integer ;
                                              D : integer ;
                                        End ;
```

Với thí dụ ở trên, ta có :

		Trái	Phải
1			
2	d	0	8
3	a	5	6
4	e	0	9
5	b	2	0
6	c	4	0
7			
8	f	0	0
9	g	0	0
10			

❖ **SỬ DỤNG CON TRỎ**. Có thể định nghĩa kiểu dữ liệu như sau :

```
Type   Pt   =   ^nut ;
       nut  =   Record
                G    : Pt ;
                     Val : t ;
                     D    : Pt ;
            End ;
```

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.3.3. DUYỆT MỘT CÂY NHỊ PHÂN.

Có 3 cách duyệt một cây nhị phân (phụ thuộc theo gốc).

1. **THỨ TỰ TRƯỚC (PREFIXÉ).**
   - Xử lý gốc.
   - Duyệt cây con trái.
   - Duyệt cây con phải.

2. **THỨ TỰ GIỮA (INFIXÉ).**
   - Duyệt cây con trái.
   - Xử lý gốc.
   - Duyệt cây con phải.

3. **THỨ TỰ SAU (POSTFIXÉ).**
   - Duyệt cây con trái.
   - Duyệt cây con phải.
   - Xử lý gốc.

THÍ DỤ. Theo cây ở thí dụ trên , ta có :

- **Trước** : a b d f c e g.
- **Giửa** : d f b a e g c.
- **Sau** : f d b g e c a.

2.4 CÂY PHỦ.

2.4.1. ĐỊNH NGHĨA.

Cho một đồ thị vô hướng G. Một cây H gọi là cây phủ của G nếu H là cây riêng phần của G chứa mọi đỉnh của G.

2.4.2. ĐỊNH LÝ.

Đồ thị G có cây phủ nếu và chỉ nếu G liên thông.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.4.3. GIẢI THUẬT TÌM CÂY PHỦ.

Xét một đồ thị G.

GIẢI THUẬT.

- **Bước 1.** Chọn tùy ý một đỉnh của G đặt vào H.
- **Bước 2.** Nếu mọi đỉnh của G đều nằm trong H thì dừng.
- **Bước 3.** Nếu không, tìm một đỉnh của G không nằm trong H mà nó có thể nối nó với một đỉnh của H bằng một cạnh. Thêm đỉnh và cạnh này vào H. Quay về bước 2.

THÍ DỤ . Cho đồ thị G theo hình vẽ sau :

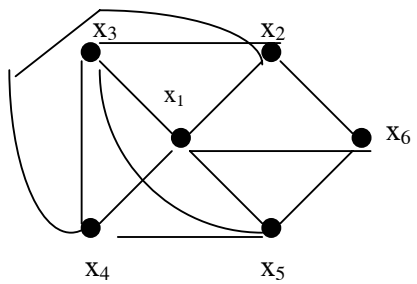


FIG. 2.3.

- Khởi từ x_1. T= ∅.
- Bước 1. Chọn x_2, T = $\{(x_1,x_2)\}$.
- Bước 2. Chọn x_3, T = $\{(x_1,x_2), (x_2,x_3)\}$.
- Bước 3. Chọn x_4, T = $\{(x_1,x_2), (x_2,x_3), (x_3,x_4)\}$.
- Bước 4. Chọn x_5, T = $\{(x_1,x_2), (x_2,x_3), (x_3,x_4), (x_4,x_5)\}$.
- Bước 5. Chọn x_6, T = $\{(x_1,x_2), (x_2,x_3), (x_3,x_4), (x_4,x_5), (x_5,x_6)\}$.

Kết quả : T là cây phủ của G .

2.4.4. ĐỊNH LÝ.

Coi một cây phủ H của G.

Thêm vào H một cạnh của G (không thuộc H), ta được một chu trình trong H. Hũy một cạnh bất kỳ trên chu trình này ra khỏi H, ta nhận được một cây phủ mới của G.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.4.5. GIẢI THUẬT KIỂM TRA TÍNH LIÊN THÔNG.

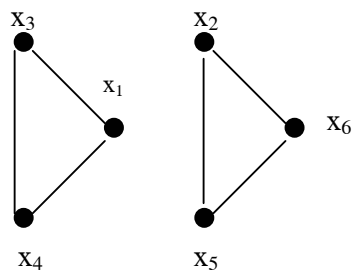
Xét một đồ thị không định hướng G.

Áp dụng giải thuật trên vào G. Khi giải thuật dừng.

- Nếu H chứa mọi đỉnh của G thì G liên thông và H là một cây phủ của G.
- Nếu H không chứa mọi đỉnh của G thì G không liên thông và H là một cây phủ của một thành phần liên thông của G.

THÍ DỤ 1. Trường hợp đồ thị G ở hình FIG. 2.3. thì ta có G liên thông.

THÍ DỤ 2. Cho đồ thị G theo hình vẽ sau :



- Khởi từ x_1. T= ∅.
- Bước 1. Chọn x_3, T = $\{(x_1,x_3)\}$.
- Bước 2. Chọn x_4, T = $\{(x_1,x_3), (x_3,x_4)\}$.

Thuật toán dừng. T là cây phủ của một thành phần liên thông của G mà thôi.

2.4.6. GIẢI THUẬT TÌM THÀNH PHẦN LIÊN THÔNG THEO CÁCH DUYỆ T THEO CHIỀU SÂU.

Do thủ tục duyệt theo chiều sâu PROF(s) cho phép thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với đỉnh s, nên số thành phần liên thông của đồ thị chính bằng số lần gọi đến thủ tục này. Vấn đề còn lại là cách ghi nhận các đỉnh trong từng thành phần liên thông bằng cách cải tiến thủ tục chiều theo chiều sâu PROF(s) như sau :

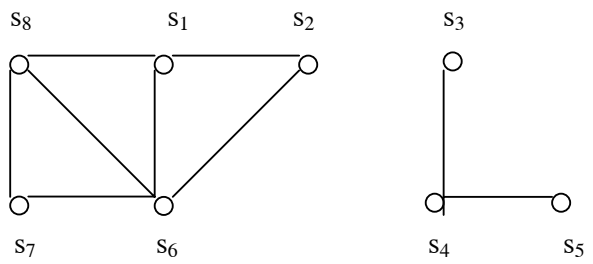
THỦ TỤC DFS(int k) ;

```
//Duyệt theo chiều sâu bắt đầu từ đỉnh k
{
    Mark[k] = socomp;
    For (int i = 1;i ≤ n ;i++)
        if (a[i][k]==1 && (Mark[i]= =0) DFS(i);
}
```

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com


```
THỦ TỤC CONNEXE ;
    {
        // Khởi tạo số liệu ban đầu cho Mark (các đỉnh đã duyệt rồi) và socomp (số
        thành phần liên thông
        For  (int  j= 1 ;j≤ n ;j++) { Mark[j] =0 ; Socomp =0 ;}
        //Gọi thủ tục để xác định các thành phần liên thông
        For  (int  i= 1 ;i≤n ;i++)
           If  Mark [i] = =0
           {
               Socomp = Socomp +1 ;
              DFS(i) ;
           }
           //In kết quả
    }
```

THÍ DỤ.



- Khởi từ s_1 . Gọi DFS(1) , ta có Tập đánh dấu $\{s_1, s_2, s_6, s_7, s_8\}$.
- i= 3 Gọi DFS(3) , ta có Tập đánh dấu $\{s_3, s_4, s_5\}$.

- **Kết quả.** Có 2 thành phần liên thông.

  $\mathbf{C_1} = \{s_1, s_2, s_6, s_7, s_8\}$.
  $\mathbf{C_2} = \{s_3, s_4, s_5\}$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.5 CÂY PHỦ TỐI THIỂU.

BÀI TOÁN 1. Cho một đồ thị liên thông G = (X,U), và,với mọi cạnh u liên kết với một con sô l(u) mà ta gọi là chiều dài (trong lượng). Vấn đề đặt ra là tìm một cây riêng phần H=(X,V) của G sao cho tổng chiều dài $\sum_u l(u)$ là nhỏ nhất.

THÍ DỤ. Bài toán này thường gặp trong viễn thông và trong nhiều trường hợp khác. Chẳng hạn, bài toán đặt ra cho chúng ta là Tìm đường dây cáp ngắn nhất để nối n thành phố lại với nhau ? Các thành phố được biểu diễn là đỉnh của một đồ thị và l((x,y) là khoảng cách giữa thành phố x và y. Mạng dây cáp nối bắt buộc phải liên thông. Ở đây, vấn đề là tìm cây riêng phần có tổng chiều dài nhỏ nhất nối tất cả các đỉnh ?

BỔ ĐỀ. Nếu G = (X,U) là một đồ thị đầy đủ và nếu tất cả các chiều dài l(u) tương ứng của các cạnh đều phân biệt thì khi ấy, Bài toán 1 có một lời giải duy nhất (X, V). Tập $V=\{v_1,v_2,\ldots,v_{n-1}\}$ nhận được theo cách sau đây :

- Chọn v_1 là cạnh có chiều dài nhỏ nhất.
- v_2 là cạnh có chiền dài nhỏ nhất sao cho $v_2 \neq v_1$ và $V_2 = \{v_1,v_2\}$ không chứa chu trình.
- v_3 là cạnh nhỏ nhất sao cho $v_3 \neq v_2 \neq v_1$ và $V_3 = \{v_1,v_2,v_3\}$ không chứa chu trình.
- Cứ thế, tiếp tục.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.5.1. THUẬT TOÁN PRIM .

Ký hiệu :

- A = Ma trận kề biểu diễn đồ thị, có trọng lượng, được định nghĩa như sau :

$$A = [a_{i,j}] = \begin{cases} l(i,j) = \text{chiều dài của cạnh cung ứng } u=(i,j) \in U \\ \propto \qquad u=(i,j) \notin U \\ 0, \qquad i=j \end{cases}$$

- M = Tập đỉnh chưa đánh dấu (có số phần tử là n_0).
- Pr(p) = Đỉnh trước đỉnh p.
- d = Tập chiều dài của Cây phủ có chiê&u dài ngắn nhất.
- Mark = Tập đỉnh đã đánh dấu (đã xét rồi), định nghĩa như sau :

$$Mark[i] = \begin{cases} 1, \text{ nếu đỉnh đã xét rồi,} \\ 0, \text{ ngược lại.} \end{cases}$$

NGUYÊN LÝ THUẬT TOÁN.

1. Khởi tạo : Xuất phát từ đỉnh 1. T = ∅,

M = {2,..n}
Pr = [1,1,…1]
d = a[1,j], j=1..n (Dòng đầu của ma trận kề A)
Mark = [1,0…0]

2. Ở mỗi bước lặp, chọn đỉnh đánh dấu là đỉnh có độ dài ngắn nhất.
   - k = $Argmin_{x \in M}$ d[x].
   - Mark[k]=1.
   - Cập nhật lại d[i], Pr[i] với i∈ M \{k} theo công thức:
     - d[i] = a[k,i] nếu d[i] > a[k,i].
     - Pr[i] = k.
   - Thay M := M\{k}.

Nếu M = ∅. Dừng. Nếu không , quay lại 2.

Độ phức tạp : O(m log n).

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

THÍ DỤ. Ta có Ma trận kề A, biểu diễn Đồ thị ở FIG. 2.3., như sau :

$$A = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 2 & 3 & 11 & 5 & 8 \\ 2 & 2 & 0 & 1 & 10 & \infty & 9 \\ 3 & 3 & 1 & 0 & 6 & 12 & \infty \\ 4 & 11 & 10 & 6 & 0 & 4 & \infty \\ 5 & 5 & \infty & 12 & 4 & 0 & 7 \\ 6 & 8 & 9 & \infty & \infty & 7 & 0 \end{array}$$

Các bước của thuật toán thực hiện như sau :

- Gán ban đầu cho : M, d, Pr :
  M = { 2, 3, 4, 5, 6}
  d = [0, 2, 3, 11, 5, 8]
  Pr = [1, 1, 1, 1, 1, 1]

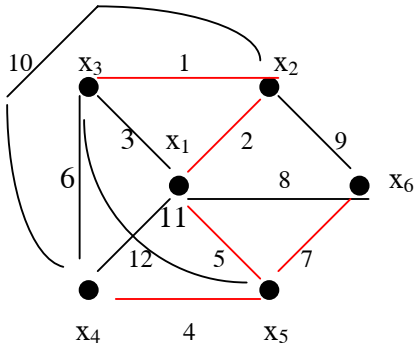
- **Bước 1.** Chọn đỉnh s_2 . Cập nhật M, d, Pr :
  M = { , 3, 4, 5, 6}
  d = [0, 2, 1, 10, 5, 8]
  Pr = [1, 1, 2 2, 1, 1]

- **Bước 2.** Chọn đỉnh s_3 . Cập nhật M, d, Pr :
  M = { , , 4, 5, 6}
  d = [0, 2, 1, 6, 5, 8]
  Pr = [1, 1, 2, 3, 1, 1]
- **Bước 3.** Chọn đỉnh s_5 . Cập nhật M, d, Pr :
  M = { , , 4, , 6}
  d = [0, 2, 1, 4, 5, 7]
  Pr = [1, 1, 2, 5, 1, 5]
- **Bước 4.** Chọn đỉnh s_4 . Cập nhật M, d, Pr :
  M = { , , , , 6}
  d = [0, 2, 1, 4, 5, 7]
  Pr = [1, 1, 2, 5, 1, 5]

Ta có Kết quả như sau :

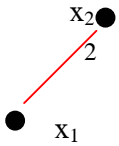
- Cây Phủ có độ dài ngắn nhất theo các Bước lặp :
  T= (x_1, x_2), (x_2,x_3),), (x_1,x_5) ,), (x_5,x_4), (x_5,x_6)} và có độ dài l(T) = 19
- Cây Phủ có độ dài ngắn nhất đọc kết quả theo d và Pr :
  T= (x_1, x_2), (x_2,x_3),), (x_5,x_4), (x_1,x_5) ,), (x_5,x_6)} và có độ dài l(T) = 19

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

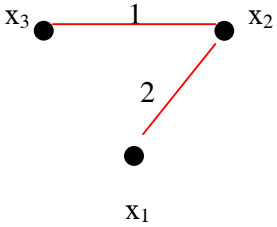


Cây ban đầu

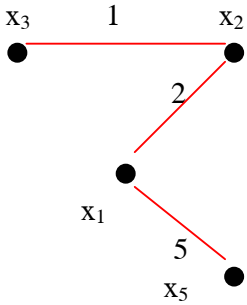
Cây khởi đầu



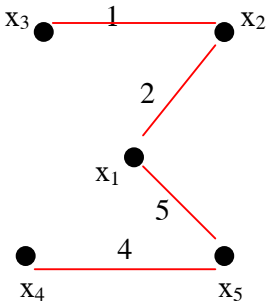
Cạnh thêm vào thứ 1



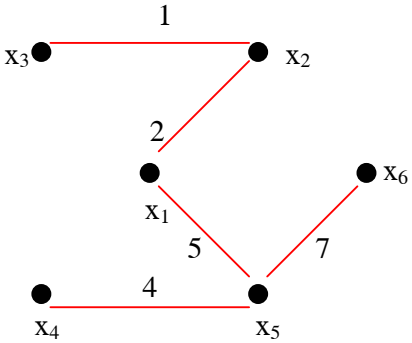
Cạnh thêm vào thứ 2



Cạnh thêm vào thứ 3



Cạnh thêm vào thứ 4



Cạnh thêm vào thứ 5.

FIG. 2.3. Tìm Cây phủ có độ dài ngắn nhất theo PRIM (s=1).

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

2.5.2. THUẬT TOÁN KRUSKAL (1956).

Cho đồ thị G = (X, U) là đồ thị liên thông không định hướng, có trọng lượng. Giả Sử đã sắp xếp các cạnh của đồ thị theo thứ tự không giảm theo chiều dài.

Ý tưởng của thuật toán KRUSKAL ở mỗi bước lặp, ta bổ sung vào tập cạnh của cây phủ H =(X, T) sao cho không tạo thành chu trình.

Thuật toán dừng khi tất cả các đỉnh của đồ thị đều được nối, nghĩa là số cạnh của H bằng n – 1. Đây là thuật toán « háu ăn », theo nghĩa là ở mỗi bước, ta chọn một lời giải tối ưu địa phương và mong muốn lời giải tối ưu địa phương này là tối ưu toàn cục.

Cây nhận được là duy nhất nếu tất cả các cạnh có chiều dài khác nhau.

Độ phức tạp : O(m log m).

THỦ TỤC KRUSKAL ;

```
Begin
  T := {∅} ;
  While Card(T) < (n-1) and (U ≠ ∅) Do Begin
      Chon u là cạnh có độ dài nhỏ nhất trong U ;
      U := U\{u} ;
     If (T ∪ {u}) không chứa chu trình) then T := T∪ {u} ;
   End ;
   If (Card(T) < n-1 ) Then Đồ thị không liên thông.
End ;
```

THÍ DỤ 1. Xem hình FIG. 2.3. Ta có :

U={(x_2, x_3),(x_1,x_2),(x_1,x_3),(x_4,x_5),(x_1,x_5),(x_3,x_4), (x_5,x_6),(x_1,x_6),(x_2,x_6),(x_2,x_4),(x_1,x_4),(x_3,x_5)}

L(U) = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Các bước của thuật toán thực hiện như sau :

- Bước 1. T= {(x_2, x_3)},
  L(T) = { 1}
- Bước 2. T= {(x_2, x_3),(x_1,x_2)},
  L(T) = { 1, 2 }
- Bước 3. T= {(x_2, x_3),(x_1,x_2),),(x_4,x_5)},
  L(T) = { 1, 2 , 4 }
- Bước 4. T= {(x_2, x_3),(x_1,x_2),),(x_4,x_5) ,(x_1,x_5)},
  L(T) = { 1, 2 , 4, 5 }
- Bước 5. T= {(x_2, x_3), (x_1,x_2),),(x_4,x_5) ,(x_1,x_5) , (x_5,x_6)}
  Kết thúc vì Card(T) = 5 = 6 (đỉnh) –1. Tổng chiều dài nhỏ nhất = 19.

Chú ý. Trong thí dụ này, ta tìm lại cây phủ giống như trong thuật toán PRIM. Nhưng, trong trường hợp tổng quát, ta có thể tìm thấy một cây phủ khác nhưng có cùng tổng trọng lượng.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

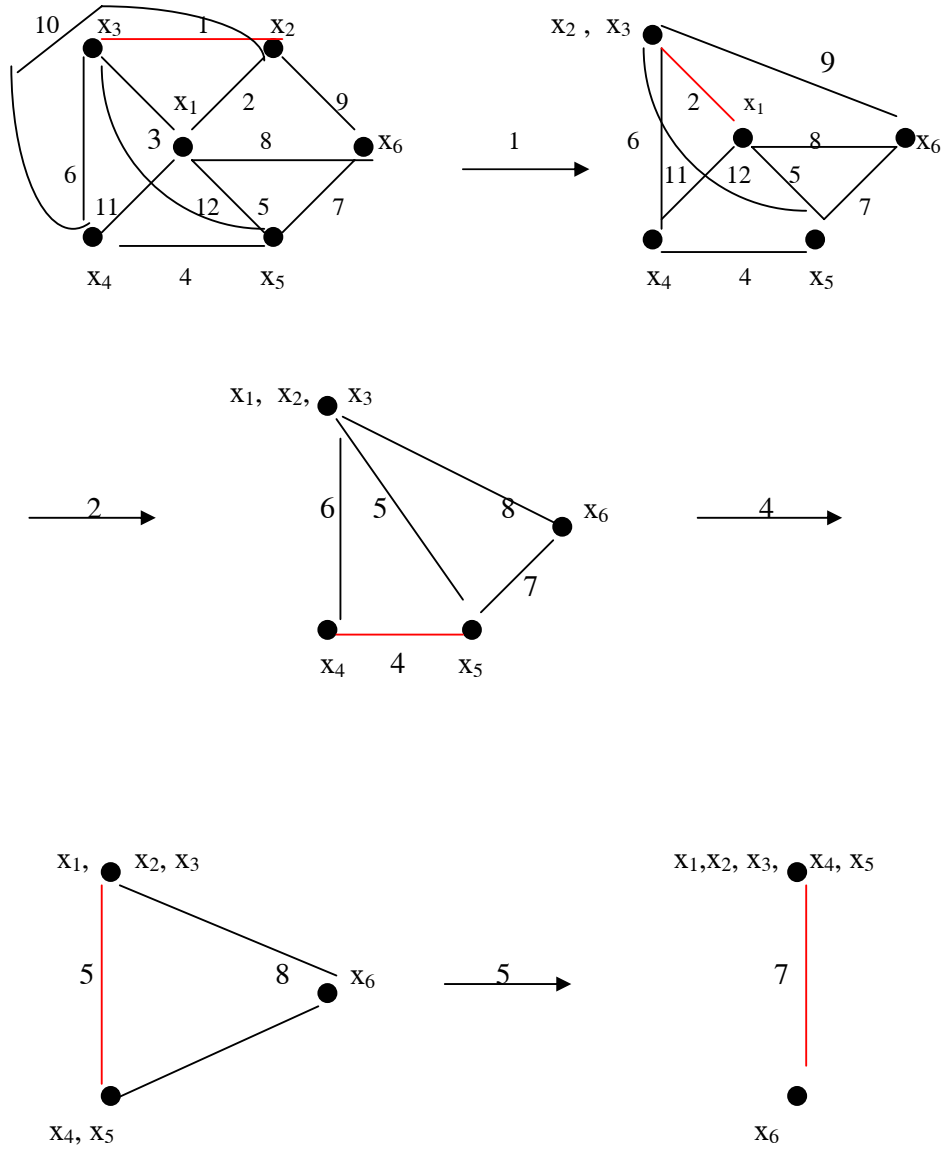


FIG. 2.4. Tìm cây phủ có chiều dài ngắn nhất theo thuật toán KRUSKAL.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

CHƯƠNG 3.

BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT.

Những bài toán tìm đường đi trong các đồ thị (đặc biệt là tìm đường đi ngắn nhất) được kể là một trong những bài toán kinh điễn, cổ trong lý thuyết đồ thị và có nhiều ứng dụng nhất.

3.1. ĐỊNH NGHĨA.

Cho G = (X, U) là một đồ thị có định giá; tương ứng với mỗi cung u=(i, j), có một chiều dài (hay trọng lượng) l(u) hay l_{ij}.
Bài toán tìm đường đi ngắn nhất giữa i và j là tìm một đường μ(i, j) từ i đến j sao cho :

$$l(\mu) = \sum_{u} l(u)$$

là ngắn nhất.

Diễn giải l(μ) : Chi chí vận chuyễn, Chi phí xây dựng, thời gian cần thiết để đi khắp,...

CHÚ Ý. Bài toán tìm đường đi ngắn nhất tương tự với bài toán tìm đường đi dài nhất.

Những thuật toán khác nhau theo những tính chất sau đây :

- $l(u) \geq 0, \forall u \in U$.
- l(u) bằng nhau $\Leftrightarrow$ $l(u) = 1, \forall u \in U$.(Bài toán đường đi ngắn nhất theo số cung)
- G không có chu trình.
- G và l(u) bất kỳ.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Và loại bài toán sau được xét :

- Tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại,
- Tìm đường đi ngắn nhất giữa các cặp đỉnh.

3.2. NGUYÊN LÝ TỐI ƯU.

Nguyên lý tối ưu phát biểu theo sự kiện là tập đường đi con của tập đường đi ngắn nhất là những đường ngắn nhất.

BỔ ĐỀ.

Xét đồ thị G = (X,U) và một hàm trọng lượng l : X x X → R, Cho C = « $x_1, x_2,...,x_k$ » là đường đi ngắn nhất từ x_1 đến x_k và với mọi (i, j) sao cho $1 \le i \le j \le k$, Cho C_{ij} = « $x_i, x_{i+1},...,x_j$ » là đường con của C từ x_i đến x_j. Khi ấy C_{ij} là một đường ngắn nhất từ x_i đến x_j.

Nguyên lý của những thuật toán tìm đường đi ngắn nhất :

- Một khoảng cách d(i) tương ứng với đỉnh x_i.
- Ở cuối thuật toán, khoảng cách này biểu diễn chiều dài ngắn nhất từ gốc đến đỉnh đang xét.

3.3. CÁC DẠNG CỦA BÀI TOÁN: TỪ MỘT ĐỈNH ĐẾN CÁC ĐỈNH CÒN LẠI.

Bài toán này còn được gọi là bài toán tìm đường đi ngắn nhất từ gốc duy nhất. Nhiều bài toán khác cũng có thể dùng thuật toán này để giải :

- Đường đi ngắn nhất đến đích duy nhất.
- Đường đi ngắn nhất từ cặp đỉnh cho trước.
- Đường đi ngắn nhất cho mọi cặp đỉnh (thuật toán gốc duy nhất từ mỗi đỉnh).

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

3.3.1. THUẬT TOÁN DIJKSTRA-MOORE (1959).

Giả thiết là các cạnh (cung) ($l(u) \geq 0$). Giả sử G có n đỉnh đánh số thứ tự từ 1 tới n. Bài toán đặt ra là tìm đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại trong đồ thị.

Ký hiệu :

- n_0 = số phần tử chưa chọn;
- A = Ma trận kề biểu diễn đồ thị, có trọng lượng, được định nghĩa như sau :

$$A = [a_{i,j}] = \begin{cases} l(i,j) = \text{chiều dài của cạnh cung ứng } u=(i,j) \in U \\ \infty & u=(i,j) \notin U \\ 0, & i=j \end{cases}$$

- Pr(p) = đỉnh trước đỉnh p theo đường đi ngắn nhất từ gốc đến đỉnh p.
- d = khoảng cách ngắn nhất từ gốc đến các đỉnh còn lại trong đồ thị. Qui ước ∞ cho các đỉnh không có đường đi từ gốc đến nó.
- Mark = Tập đỉnh đã đánh dấu (đã xét rồi), định nghĩa như sau :

$$Mark[i] = \begin{cases} 1, \text{ nếu đỉnh đã xét rồi,} \\ 0, \text{ ngược lại.} \end{cases}$$

NGUYÊN LÝ THUẬT TOÁN.

1. **Khởi tạo** : Xuất phát từ đỉnh 1 ; $n_0 = n - 1$:

   Pr = [1,1,...1]
   d = a[1,j], j=1..n (Dòng đầu của ma trận kề A)
   Mark = [1,0...0]

2. **Ở mỗi bước lặp**, chọn đỉnh đánh dấu là đỉnh có độ dài ngắn nhất trong những đỉnh chưa đánh dấu, nghĩa là chọn đỉnh **k** sao cho :
   - d[k] = Min {d[i] : Mark[i]= 0 } ;
   - Mark[k]=1.
   - Cập nhật lại d[j], Pr[j] với những đỉnh j chưa đánh dấu (Mark[j]=0) theo công thức:
     - d[j] = d[k] + a[k,j] nếu d[j] > d[k] +a[k,j].
     - Pr[j] = k.

Nếu tất cả mọi đỉnh đã được chọn, nghĩa là $n_0 = 0$. Dừng. Nếu không , quay lại 2.

THỦ TỤC DIJKSTRA – MOORE ;

```
❖ //Giả sử đã nhập ma trận chiều dài 1 theo dạng ma trận kề A
❖ //Gán ban đầu cho d, Pr, Mark, n0.
  For (int j= 1; j≤ n ; j++)   { d[j] = a(1,j) ; pr[j]=1 ; Mark[j] = 0;}
  Mark[1] =1 ; n0 = n-1 ;
❖ WHILE (n0 > 0)
  {d[k] = Min {d[j] : Mark[j]= 0 } ;
  // Cập nhật lại n0, d và Pr, Mark
    Mark[k] =1 ; n0 = n0 - 1 ;
    For (int j= 1; j≤ n ; j++)  if (Mark [j] = 0) && (d[k]+ a[k,j] < d[j])
      { d[j] = d[k] +a[k,j] ; pr[j]=k}
  }
```

Độ phức tạp : $O(n^2)$ hay O(mlogn)

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

THÍ DỤ.

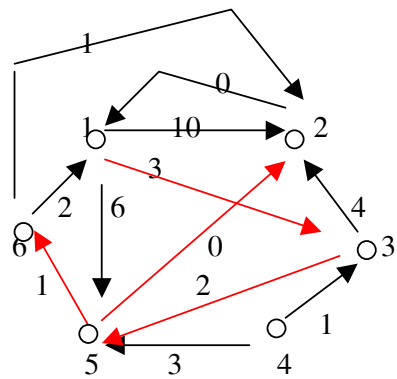


FIG.3.1. Đồ thị có định hướng, có trọng lượng.

Ma trận kề A :

$$A = \begin{array}{c|cccccc|} & 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 10 & 3 & \propto & 6 & \propto \\ 2 & 0 & \propto & \propto & \propto & \propto & \propto \\ 3 & \propto & 4 & 0 & \propto & 2 & \propto \\ 4 & \propto & \propto & 1 & 0 & 3 & \propto \\ 5 & \propto & 0 & \propto & \propto & 0 & 1 \\ 6 & 2 & 1 & \propto & \propto & \propto & 0 \end{array}$$

- **Gán Ban đầu.** Cho Mark, d, Pr :
  Mark = [1, 0, 0, 0, 0, 0]
  d = [0, 10, 3, ∝, 6, ∝]
  Pr = [1, 1, 1, 1, 1, 1]
- **Bước 1.** Chọn đỉnh s_3. Cập nhật Mark, d, Pr :
  Mark = [1, 0, 1, 0, 0, 0]
  d = [0, 7, **3**, ∝, 5, ∝]
  Pr = [1, 3, **1**, 1, 3, 1]
- **Bước 2 .** Đỉnh hiện thời là s_3. Chọn đỉnh s_5. Cập nhật Mark, d, Pr :
  Mark = [1, 0, 1, 0, 1, 0]
  d = [0, 5, **3**, ∝, 5, 6]
  Pr = [1, 5, **1**, 1, 3, 5]
- **Bước 3 .** Đỉnh hiện thời là s_5 . Chọn đỉnh s_2. Cập nhật Mark, d, Pr :
  Mark = [1, 1, 1, 0, 1, 0]
  d = [0, 5, **3**, ∝, 5, 6]
  Pr = [1, 5, **1**, 1, 3, 5]
- **Bước 4 .** Đỉnh hiện thời là s_2 . Chọn đỉnh s_6. Cập nhật Mark, d, Pr :
  Mark = [1, 1, 1, 0, 1, 1]
  d = [0, 5, **3**, ∝, 5, 6]
  Pr = [1, 5, **1**, 1, 3, 5]

Thuật toán kết thúc vì đỉnh s_4, ta có $d[s_4]$ = Min {d[j] : Mark[j]= 0}= $d[s_4]$ = ∝.
Từ thuật toán , ta có kết quả sau :

d = [0, 5, **3**, ∝, 5, 6]
Pr = [1, 5, **1**, 1, 3, 5]

- Đường đi ngắn nhất từ s_1 đến s_2 : $s_1 \to s_3 \to s_5 \to s_2$ và độ dài là **5**
- Đường đi ngắn nhất từ s_1 đến s_3 : $s_1 \to s_3$ và độ dài là **3**
- Đường đi ngắn nhất từ s_1 đến s_5 : $s_1 \to s_3 \to s_5$ và độ dài là **5**
- Đường đi ngắn nhất từ s_1 đến s_6 : $s_1 \to s_5 \to s_6$ và độ dài là **6**
- Không có đường đi ngắn nhất từ đỉnh s_1 đến s_4 ($d[s_4]$ = ∝) , vì không có đường nối từ s_1 đến s_4.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

GHI CHÚ.

Giả thiết « Hàm trọng lượng không âm » là bắt buộc. Chẳng hạn, sử dụng thuật toán Dijktra-Moore cho đồ thị ở hình FIG.3.2, dẫn đến kết quả sai nếu ta chọn gốc là đỉnh s_1. Thật vậy, đầu tiên, ta chọn đỉnh s_2, ($s_1 \rightarrow s_2$) trong khi đó, đường đi ngắn nhất là đường đi từ đỉnh s_1 đến s_2 qua s_3.

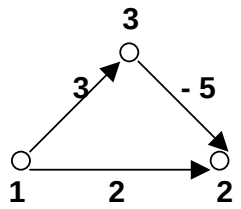


FIG. 3.2. Đồ thị có định hướng, có trọng lượng bất kỳ.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

3.3.2. THUẬT TOÁN BELLMAN-FORD (1958-1962)

Sự hiện diện của dấu bất kỳ của trọng lượng (hay chiều dài) cho phép, chẳng hạn, có thể cải tiến chi phí hay lợi nhuận. Thuật toán DIJKSTRA-MOORE không cho phép xét tới những cạnh (cung) có trọng lượng không âm, vì trong trường hợp một cạnh được đánh dấu, thì ta không thể thay đổi gì cho những bước lặp tiếp theo. Thuật toán DIJKSTRA-MOORE còn được gọi là **gán nhãn cố định .**

Để giải quyết cho trường hợp đồ thị có trọng lượng bất kỳ, ta một xét thuật toán cho phép một đánh dấu chỉ được xác định hoàn toàn khi thuật toán kết thúc. Một kiểu thuật toán như vậy được gọi là **điều chỉnh nhãn.**

Thuật toán **BELLMAN-FORD** chỉ có giá trị cho các đồ thị không có chu trình, có trọng lượng bất kỳ.

Ký hiệu :

- Tập đỉnh được đánh số thứ tự từ 1 ..n.
- Pr(p) = đỉnh trước đỉnh p theo đường đi ngắn nhất từ gốc đến đỉnh p.
- d = khoảng cách ngắn nhất từ gốc đến các đỉnh còn lại trong đồ thị.
- Mark = Tập đỉnh đã đánh dấu (đã xét rồi), định nghĩa như sau :

$$\text{Mark}[i] = \begin{cases} 1, \text{ nếu đỉnh đã xét rồi,} \\ 0, \text{ ngược lại.} \end{cases}$$

Khoảng cách ngắn nhất từ gốc đến một đỉnh v chỉ được tính khi tất cả các phần tử trước của v ($\Gamma^-(v)$) đã được đánh dấu rồi. Một đỉnh bất kỳ, khi chưa đánh dấu, thì khoảng cách từ gốc đến đỉnh đó chưa biết (chưa tính).

NGUYÊN LÝ THUẬT TOÁN

1. Gán các giá trị ban đầu.

- Chọn đỉnh s_1 làm gốc.
- Mark = [1,0...0] ; d[1] = 0 ; Pr[1] = 1.

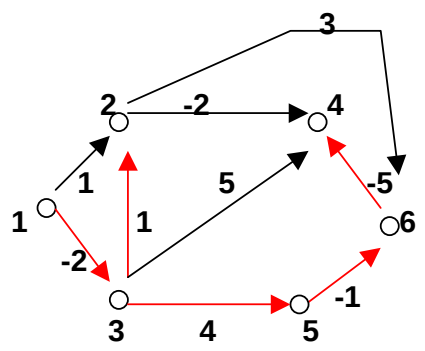
2. Ở mỗi bước lặp :

- Chọn đỉnh **k** chưa đánh dấu sao cho tất cả đỉnh trước của k đã đánh dấu rồi , nghĩa là : **Mark[k] = 0 và $\forall$ j $\in$ Γ^-(k) : Mark[j]= 0**
- Cập nhật Mark : Mark[k] =1 ;
- Tính d[k] = min { d[i] + a[i, k]: i $\in$ Γ^-(k)}, và Pr[k] là chỉ số đạt min.

ĐỘ PHỨC TẠP : O(nm). $O(n^3)$ Cho các đồ thị dầy, i.e., những đồ thị mà $m \approx n^2$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

THÍ DỤ.



Gán ban đầu : Mark, d, Pr :
Mark = [1, 0, 0, 0, 0, 0}**,**
d[1] = 0 ;
Pr [1] = 1
$\Gamma^-(2)=\{1,3\}$;$\Gamma^-(3)=\{1\}$;$\Gamma^-(4)=\{2,3,6\}$
$\Gamma^-(5)=\{3\}$; $\Gamma^-(6)=\{2,5\}$

FIG.3.1. Đồ thị có định hướng, có trọng lượng bất kỳ, không có chu trình, gốc đỉnh 1.

- **Bước 1.** Chọn đỉnh **3** vì Γ^-**(3)={1}** . Cập nhật Mark[3], Tính d[3] và Pr[3] :
  Mark[3] = 1 ; d[3] = -2 ; Pr[3] = 1;

- **Bước 2.** Ở bước lặp này, ta có thể chọn đỉnh 5 (hay đỉnh 2).
  Cập nhật Mark[5], Tính d[5] và Pr[5] :
  Mark[5] = 1 ; d[5] = 2 ; Pr[5] = 3;

- **Bước 3.** Chọn đỉnh **2** . Cập nhật Mark[2], Tính d[2] và Pr[2] :
  Mark[2] = 1 ; d[2] = -1 ; Pr[2] = 3;

- **Bước 4.** Chọn đỉnh **6** . Cập nhật Mark[6], Tính d[6] và Pr[6] :
  Mark[6] = 1 ; d[6] = 1; Pr[6] = 5

- **Bước 5.** Chọn đỉnh **4** . Cập nhật Mark[4], Tính d[4] và Pr[4] :
  Mark[4] = 1 ; d[4] = - 4 ; Pr[4] = 6

Thuật toán kết thúc vì tất cả các đỉnh đã được chọn rồi.

Từ thuật toán , ta có kết quả sau :

d = [0, -1, -2, -4, 2, 1]
Pr = [1, 3, 1, 6, 3, 5]

- Đường đi ngắn nhất từ s_1 đến s_2 : $s_1 \to s_3 \to s_2$ và độ dài là **-1**
- Đường đi ngắn nhất từ s_1 đến s_3 : $s_1 \to s_3$ và độ dài là **-2**
- Đường đi ngắn nhất từ s_1 đến s_4 : $s_1 \to s_3 \to s_5 \to s_6 \to s_4$ và độ dài là - 4
- Đường đi ngắn nhất từ s_1 đến s_5 : $s_1 \to s_3 \to s_5$ và độ dài là **2**
- Đường đi ngắn nhất từ s_1 đến s_6 : $s_1 \to s_3 \to s_5 \to s_6$ và độ dài là 1

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

3.4. GIỮA TẤT CẢ CÁC CẶP ĐỈNH: THUẬT TOÁN FLOYD (1962).

Ta sẽ tính một ma trận khoảng cách n x n. Nếu tất cả chiều dài không âm ($l(u) \geq 0$) ta có thể áp dụng n lần thuật toán Dijktra-Moore cho mỗi đỉnh i. . Nếu đồ thị có chứa chiều dài âm ($l(u) < 0$) ta có thể áp dụng n lần thuật toán Bellman-Ford cho mỗiđỉnh i. Thuật toán Floyd có cách tiếp cận khác có lợi cho trường hợp ma trận dầy.

Ký hiệu :

- A : ma trận trọng lượng, được gán giá trị ban đầu như sau :

$$A[i,j] = \begin{cases} 0 & \text{nếu } i = j \\ l(i, j) & \text{nếu } (i, j) \in U \\ \infty & \text{ngược lại.} \end{cases}$$

- P : ma trận các đỉnh trước, được gán giá trị ban đầu như sau :
  P[i,j] = i, trong đó P[i,j] là đỉnh trước của đỉnh j trên đường đi từ gốc i đến j

Khi kết thúc thuật toán, ta có :

P[i,j] = đỉnh trước của j trên đường đi ngắn nhất từ gốc i đến đỉnh j, với chiều dài tương ứng là A[i,j].

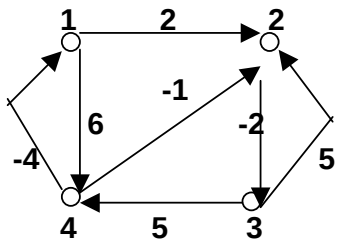
THỦ TỤC FLOYD(L, P)

```
For (k =1;  k≤ n ; k++)
    For (i =1 ;i≤ n ; i++)
        For (j =1 ;j≤ n ; j++)
            If  (a[i,k] +  a[k,j]  <  a[i,j])
            { a[i,j] := a[i,k]  +  a[k,j]  ;  p[i,j] :=p[k,j] ;}
```

Độ phức tạp : $O(n^3)$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

THÍ DỤ.



Gán ban đầu : cho các ma trận A, P.

$$A_0 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & \propto & 6 \\ 2 & \propto & 0 & -2 & \propto \\ 3 & \propto & 5 & 0 & 5 \\ 4 & -4 & -1 & \propto & 0 \end{array} \qquad P_0 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{array}$$

Các bước lặp :

- **k =1.**

$$A_1 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & \propto & 6 \\ 2 & \propto & 0 & -2 & \propto \\ 3 & \propto & 5 & 0 & 5 \\ 4 & -4 & -2 & \propto & 0 \end{array} \qquad P_1 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 1 & 4 & 4 \end{array}$$

- **k = 2**

$$A_2 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & 0 & 6 \\ 2 & \propto & 0 & -2 & \propto \\ 3 & \propto & 5 & 0 & 5 \\ 4 & -4 & -2 & -4 & 0 \end{array} \qquad P_2 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{array}$$

- **k =3**

$$A_3 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & 0 & 5 \\ 2 & \propto & 0 & -2 & 3 \\ 3 & \propto & 5 & 0 & 5 \\ 4 & -4 & -2 & -4 & 0 \end{array} \qquad P_3 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 3 \\ 0 & 2 & 2 & 3 \\ 0 & 3 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{array}$$

- **k = 4**

$$A_4 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 2 & 0 & 5 \\ 2 & -1 & 0 & -2 & 3 \\ 3 & 1 & 3 & 0 & 5 \\ 4 & -4 & -2 & -4 & 0 \end{array} \qquad P_4 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 3 \\ 0 & 2 & 2 & 3 \\ 4 & 1 & 3 & 3 \\ 4 & 1 & 2 & 4 \end{array}$$

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Cách nhận biết đường đi ngắn nhất.

Để nhận được đường đi ngắn nhất từ s_1 đến s_j, ta sử dụng dòng thứ i của ma trận P. Chẳng hạn, ta muốn nhận được đường đi ngắn nhất $\mu : s_4 \rightarrow s_3$, ta tham khảo ma trận P như sau : P[4,3]=2 :s_2 là đỉnh trước của s_3 ; P[4,2]=1 : s_1 là đỉnh trước của s_2 ; P[4,1]=4 :s_4 là đỉnh trước của s_1.
Cuối cùng, kết quả là $\mu = s_4 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$.

Một trong ứng dụng của Thuật toán FLOYD là tìm đường đi giữa hai đỉnh. Thuật toán này được WARSHALL phát triển cùng năm (1962), và thuật toán thường mang tên FLOYD-WARSHALL ».

Ký hiệu :

- A = ma trận kề của đồ thị, được gán giá trị ban đầu như sau :

$$A[i,j] = \begin{cases} 1 & \text{nếu } (i, j) \in U \\ 0 & \text{ngược lại.} \end{cases}$$

- P = ma trận các đỉnh trước, được gán giá trị ban đầu như sau :

$$P[i,j] = \begin{cases} 0 & \text{nếu } a[i,j] = 0, \\ 1 & \text{ngược lại.} \end{cases}$$

Khi kết thúc thuật toán :
P[i,j] = đỉnh trước của j trên đường đi từ đỉnh i đến đỉnh j (nghĩa là a[i,j]=1).

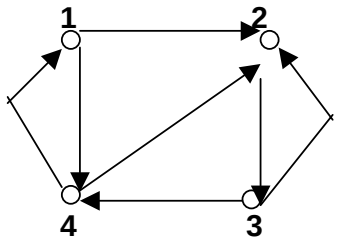
THỦ TỤC FLOYD-WARSHAL(A, P)

```
For (k =1 ;k≤ n ; k++)
    For (i =1 ;i≤ n ; i++)
        For (j =1 ;j≤ n ; j++)
            If  (a[i,j]  = =  0)
            {       a[i,j] = a[i,k] *a[k,j]  ; p[i,j] =p[k,j] }
```

Độ phức tạp : $O(n^3)$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

THÍ DỤ.



Gán ban đầu : cho các ma trận A, P.

$$A_0 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 0 \end{array} \qquad P_0 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & 3 \\ 4 & 4 & 0 & 0 \end{array}$$

Các bước lặp :

- **k =1.**

$$A_1 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 \end{array} \qquad P_1 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 0 & 3 \\ 4 & 4 & 0 & 1 \end{array}$$

- **k = 2**

$$A_2 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array} \qquad P_2 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 3 & 2 & 3 \\ 4 & 4 & 2 & 1 \end{array}$$

- **k =3**

$$A_3 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array} \qquad P_3 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 1 \\ 0 & 3 & 2 & 3 \\ 0 & 3 & 2 & 3 \\ 4 & 4 & 2 & 1 \end{array}$$

- **k = 4**

$$A_4 = \begin{array}{c|cccc|} & 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 \end{array} \qquad P_4 = \begin{array}{|cccc|} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 1 \\ 4 & 3 & 2 & 3 \\ 4 & 3 & 2 & 3 \\ 4 & 4 & 2 & 1 \end{array}$$

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

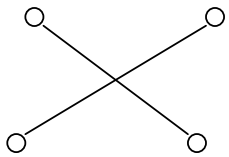
CHƯƠNG 4.

ĐỒ THỊ PHẲNG & BÀI TOÁN TÔ MÀU.

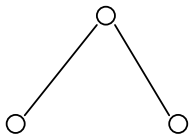
4.1. ĐỊNH NGHĨA VỀ ĐỒ THỊ PHẲNG.

Đồ thị phẳng là một đồ thị có thể biểu diễn trên một mặt phẳng (hay trên hình cầu) sao cho hai cung (hay hai cạnh) không cắt nhau.

Ghi chú. Hai cạnh có chung một đỉnh được gọi là không cắt nhau.

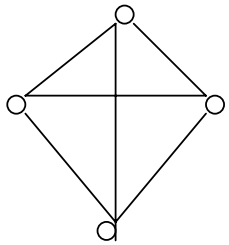


Cắt nhau

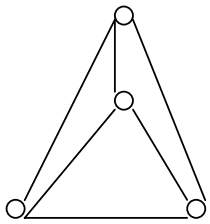


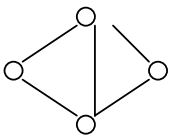
Không cắt nhau .

Thí dụ. Đồ thị G_1 là đồ thị phẳng và G_2 , G_3 là các biểu diễn phẳng của G_1.



Đồ thị G_1





Biểu diễn G_2, G_3 của G_1.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Cho G là đồ thị phẳng. Một **mặt** (FACE) của G là một miền, giới hạn bởi các cạnh, không có đỉnh lẫn cạnh ở bên trong. Trong các mặt này luôn luôn có một và chỉ một mặt vô hạn. **Đường biên** (CONTOUR) của một mặt r là chu trình hợp thành từ các cạnh biên của r. Hai mặt r và s được gọi là **KỀ** (ADJACENTES) nếu đường biên của chúng có chung ít nhất một cạnh. Hai mặt không có chung một đỉnh nào thì sẽ không kề.

THÍ DỤ.

❖ Một bản đồ địa dư là một đồ thị phẳng (với điều kiện là không có đảo). Đồ thị này đặc biệt mỗi đỉnh có bậc ≥ 3. Mặt h là mặt vô hạn, những mặt còn lại a, b, c, d, e, f, g là những mặt hữu hạn.

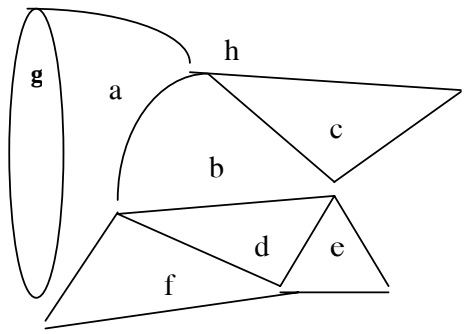


FIG. 4.1. ĐỒ THỊ PHẲNG.

❖ **Bài toán ba làng và ba nhà máy**. Ta có 3 làng a, b, c, mà ta muốn đặt đường nối với 3 nhà máy : một nhà máy cung cấp nước d, một nhà máy cung cấp ga e, một nhà máy cung cấp điện f. Vấn đề đặt ra là , ta có thể đặt trên một mặt phẳng sao cho các đường dẫn không giao nhau ngoài các đỉnh cực biên ? Đồ thị biểu diễn 3 làng và 3 nhà máy cho phép định nghĩa một lớp các đồ thị không phẳng.

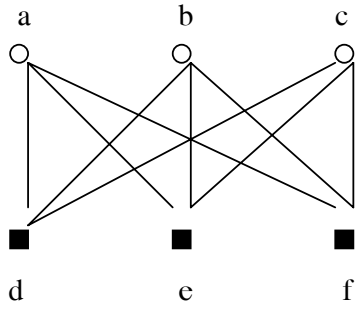


FIG 4.2. ĐỒ THỊ KHÔNG PHẲNG LOẠI 1 : $K_{3,3}$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

4.2. CÔNG THỨC EULER , HỆ QUẢ & THÍ DỤ.

4.2..1. CÔNG THỨC EULER.

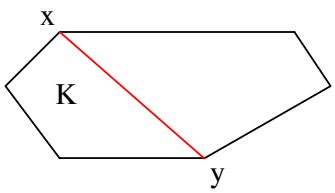
Cho một đồ thị phẳng liên thông có n đỉnh, m cạnh và f mặt, ta có

$$n - m + f = 2$$

Chứng minh. Truy chứng trên số cạnh :

- $m = 1$. Ta có $n = 2$ đỉnh và $f=1$ mặt. Ta có $n - m + f = 2 - 1 + 1 = 2$
  Vậy công thức EULER đúng cho trường hợp $m = 1$.
- Giả sử công thức EULER đúng cho trường hợp đồ thị G_{i-1} có $m_i - 1$ cạnh. Ta sẽ chứng minh công thức EULER cũng đúng cho trường hợp đồ thị có m_i cạnh.
  Gọi cạnh $u = (x,y)$ là cạnh vẽ thêm vào G_{i-1} để có G_i.
  Hiễn nhiên là có it nhất một đỉnh thuộc G_{i-1} và $u=(x,y)$ thuộc một mặt K của G_{i-1}. Giả sử $x \in G_{i-1}$. Có 2 trường hợp xãy ra :

1. $y \in K$. Do đó ta có :



$f_i = f_{i-1} + 1$.
$n_i = n_{i-1}$
$m_i = m_{i-1} + 1$
Ta có :
$$n_i - m_i + f_i = n_i - (m_{i-1} + 1) + (f_{i-1} + 1).$$
$$= n_i - m_{i-1} + f_{i-1} = 2$$
Vậy công thức EULER đúng.

2. $y \notin K$. Ta có :
$f_i = f_{i-1}$.
$n_i = n_{i-1} + 1$
$m_i = m_{i-1} + 1$
Ta có :
$$n_i - m_i + f_i = (n_i + 1) - (m_{i-1} + 1) + f_{i-1}$$
$$= n_i - m_{i-1} + f_{i-1} = 2$$
Vậy công thức EULER đúng.

Vậy công thức EULER đúng với mọi m.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

4.2.2. Hệ quả.

Trong một đồ thị đơn giản phẳng, liên thông bất kỳ có n đỉnh, m cạnh (m > 2) và f mặt. Khi ấy, ta có :

$$3f/2 \leq m \leq 3n - 6. \quad (1)$$

Chứng minh.
Mỗi mặt bị bao ít nhất 3 cạnh, mỗi cạnh thuộc 2 mặt.
Ba cạnh xác định tối đa 2 mặt. Vậy số mặt tối đa là 2m/3.
Ta có $f \leq 2m/3$. Dùng công thức EULER suy ra bất đẳng thức (1).

4.2.3. Hệ quả.

Trong tất cả các đồ thị phẳng đơn giản, có ít nhất một đỉnh có bậc ≤ 5.

Chứng minh.
Giả sử mọi đỉnh có bậc > 6. Khi ấy $2m > 6n \Rightarrow m > 3n > 3n - 6$. Mâu thuẩn.

4.2.4. THÍ DỤ.

Dùng công thức EULER, ta sẽ chứng minh là tất cả đồ thị đầy đủ 5 đỉnh K_5 là không phẳng.

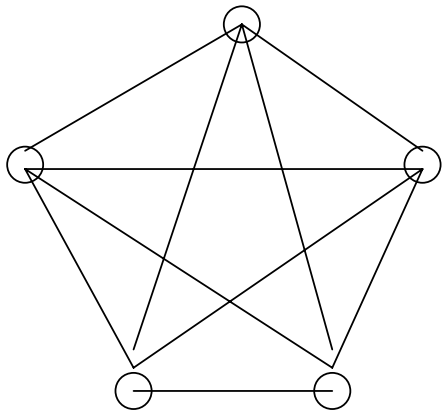


FIG 4.3. ĐỒ THỊ KHÔNG PHẲNG LOẠI 2 : K_5.

Chứng minh.
Ta có số đỉnh $n = 5$, Số cạnh $m = n(n-1)/2 = 10$.
Nếu K_5 phẳng, áp dụng hệ quả 3.2.2 ta có :

$$10 = m \leq 3n - 6 = 3x5 - 6 = 9.$$

Mâu thuẩn. Vậy K_5 không phẳng.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

Nhận xét.

- Đồ thị những làng và những nhà máy (Loại 1 : $K_{3,3}$) và đồ thị đầy đủ 5 đỉnh (loại 2 :K_5) cho phép định nghĩa tất cả những đồ thị mà không phẳng.
- K_5, $K_{3,3}$ cùng là đồ thị đều.
- Đồ thị K_5 không phẳng với số đỉnh nhỏ nhất, đồ thị $K_{3,3}$ là đồ thị không phẳng có số cạnh nhỏ nhất, và cả hai là đồ thị không phẳng đơn giản nhất.

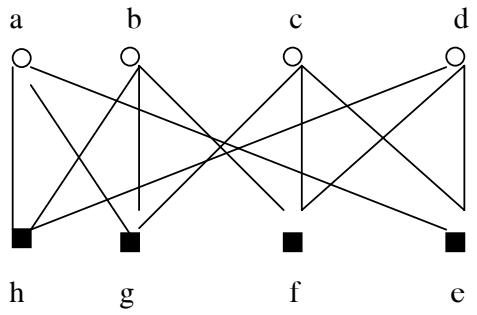
4.3. BẤT ĐẲNG THỨC CẠNH- ĐỈNH.

4.3.1. THÍ DỤ.

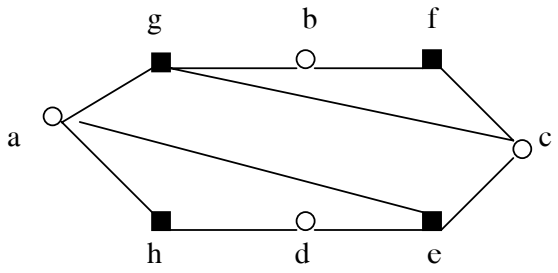
Ta xét bài toán xác định xem đồ thị G cho trước có phẳng không ?

- **THÍ DỤ 1**. Cho đồ thị K_4.. K_4 phẳng.

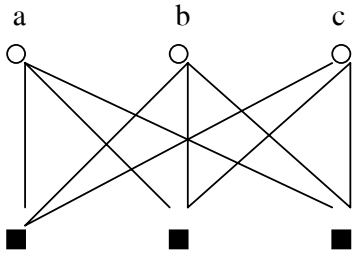
- **THÍ DỤ 2.** Cho đồ thị G sau :



G phẳng vì ta có thể vẽ lại như sau :



- **THÍ DỤ 3.** Đồ thị sau đây không phẳng.



Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

1 2 3

4.3.2. BẤT ĐẲNG THỨC CẠNH – ĐỈNH.

Cho G là một đồ thị phẳng liên thông có n đỉnh, m cạnh và đường biên của các mặt có số cạnh $g \geq 3$. Khi ấy, ta có :

$$m \leq (n-2)\, g/(g-2).$$

Chứng minh.

Giả sử ma trận kề cạnh- mặt có dạng :

$$A = \begin{array}{c|ccccc|} & f_1 & f_2 & . & f_j & f_F \\ m_1 & & & & . & \\ m_2 & & & & . & \\ . & & & & . & \\ m_I & . & . & . & m_{ij} & \\ . & & & & & \\ m_f & & & & & \end{array}$$

trong đó :

$m_{ij} =$ 1 nếu m_I là cạnh biên của f_j,
0 ngược lại.

Xét hàng thứ i, ta có :

$$\sum m_{ij} \leq 2 \quad (\text{ vì } m_{ij} \text{ là cạnh biên của nhiều nhất 2 mặt})$$

Suy ra $\sum\sum m_{ij} \leq 2m$ (1)

Xét cột thứ j, ta có :

$$\sum m_{ij} \geq g \quad (\text{vì mặt } f_j \text{ có it nhất } g \text{ cạnh biên})$$

Suy ra $\sum\sum m_{ij} \geq gf$ (2)

Theo công thức EULER, ta có :

$$n - m + f = 2 \qquad (3)$$

Theo (2), (1), ta có :

$$\begin{aligned} & gf = g(2 + m - n) && \leq 2m \\ & (2 + m - n) && \leq 2m/g \\ \Leftrightarrow\ & m(1-2/g) && \leq n - 2 \\ \Leftrightarrow\ & m && \leq (n-2)\, g/(g-2) \end{aligned}$$

BĐT đã chứng minh xong.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

❖ **THÍ DỤ.**

Nhờ Bất đẳng thức trên, ta sẽ chứng minh được rằng đồ thị 3 làng và 3 nhà máy $K_{3,3}$, xem hình FIG. 4.2. không phẳng.
Thật vậy, nhận xét rằng mọi chu trình trong $K_{3,3}$ có số cạnh ít nhất là 4. Vậy nếu $K_{3,3}$ phẳng mọi mặt phải có số cạnh ít nhất là 4.
Theo Bất đẳng thức trên, ta có : $9 = m \leq (6-2)\ 4/(4-2) = 8$. Mâu thuẩn.
Vậy $K_{3,3}$ không phẳng.

4.4. PHÉP ĐỒNG DẠNG.

4.4.1. ĐỊNH NGHĨA.

Hai đồ thị được gọi là đồng dạng với nhau nếu đồ thị này có được bằng cách biến đổi đồ thị kia theo cách thêm đỉnh bậc 2 hoặc bỏ đi đỉnh bậc 2.

THÍ DỤ. a ——— b → a —— c ——— b → a —— b

Thêm Đỉnh c bậc 2 vào ab

Bớt Đỉnh c bậc 2 khỏi acb.

4.4.2. BỔ ĐỀ.

Giả sử H là đồ thị con của G. Khi ấy :
❖ Nếu G phẳng thì H phẳng.
❖ Nếu H không phẳng thì G cũng không phẳng.

4.4.3. BỔ ĐỀ.

Mọi đồ thị là phẳng nếu đồng dạng của nó là phẳng.

4.5. ĐỊNH LÝ KURATOWSKI.

Đồ thị G là phẳng nếu và chỉ nếu G không chứa một đồ thị con đồng cấu với K_5 cũng như với $K_{3,3}$.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

4.6. BÀI TOÁN TÔ MÀU ĐỒ THỊ.

4.6.1. ĐỊNH NGHĨA.

Phép tô màu một đồ thị là phép gán màu cho các đỉnh của đồ thị sao cho hai đỉnh kề nhau có màu khác nhau.

Một cách hình thức có thể định nghĩa phép tô màu như sau :

Phép tô màu là một ánh xạ $\gamma : X \rightarrow N$ sao cho $\forall (x, y) \in X$, $\gamma(x) \neq \gamma(y)$.

THÍ DỤ.

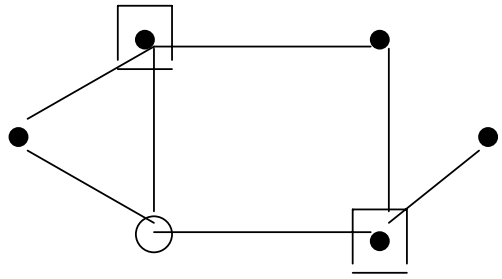


FIG 4.4.

Số màu (phân biệt) ít nhất cần thiết để tô màu các đỉnh của đồ thị G được gọi là **Sắc tố** (CHROMATIQUE) và ký hiệu là $\gamma(G)$.

Một đồ thị G $\gamma(G) \leq k$ được gọi là **k-sắc tố.**

Chận trên của sắc tố được cho bởi $d + 1$ với d bậc lớn nhất của đỉnh.

$$\gamma(G) \leq d + 1$$

4.6.2. CÁC ỨNG DỤNG.

❖ **XẾP LỊCH THI.**

Giả sử ta khảo sát việc thi vấn đáp của một kỳ thi. Có những ràng buộc sau :

- Một thầy , một lúc chỉ có thể hỏi thi một em.
- Một thí sinh thi với một thầy vào một thời gian đã định trước.

Sự phân bố thí sinh thi với thầy nào đã được ấn định trước. (Thầy P_i thí sinh E_j) :

THÍ DỤ. (P_1, E_1), (P_1, E_2), (P_1, E_3), (P_2, E_1), (P_2, E_2),

❖ **BẢN ĐỒ ĐỊA DƯ**.

Một bài toán hết sức lý thú là tô màu các bản đồ sao cho hai vùng khác nhau không cùng một màu.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

4.6.3. THUẬT TOÁN TÔ MÀU.

DỮ LIỆU : Đồ thị $G = (X, U)$.

KẾT QUẢ : Một phép tô màu $\gamma : X \rightarrow N$.

BEGIN.

Cho $\tau = x_1, x_2, ..., x_n$ là một phép đánh số thứ tự các đỉnh của G.

Cho $C = \{1, 2, ..., k\}$ tập các màu.

FOR i=1 To n Do
$\gamma(x_i) = \text{Min}\{k \in C : \forall$ đỉnh y kề $x_i, \gamma(y) \neq k\}$

END.

4.6.4. ĐỊNH LÝ.

Nếu G có chứa một đồ thị con đẳng hình với K_m thì $\gamma(G) \geq m$.

CHỨNG MINH. Hiễn nhiên.

4.6.5. ĐỊNH LÝ 5 MÀU (KEMPE-HEAWOOD).

Mọi đồ thị phẳng đều có 5-sắc tố.

Simpo PDF Merge and Split Unregistered Version - http://www.simpopdf.com

4.6.6. BÀI TOÁN 4 MÀU.

❖ **GIẢ THIẾT BÀI TOÁN 4 MÀU.**

Trên một bản đồ bất kỳ, ta nói nó được tô màu nếu mỗi miền của bản đồ được tô một màu xác định sao cho 2 miền kề nhau (chung một phần biên) phải được tô bằng hai màu khác nhau. Vấn đề đặt ra là cần dùng tối thiểu bao nhiêu màu để tô được một bản đồ bất kỳ. Vấn đề này được đặt ra từ năm 1852 do giáo sư De Morgan đặt ra : « Mọi bản đồ đều có thể tô bằng 4 màu sao cho hai nước nằm kề nhau phải được tô bằng hai màu khác nhau ». Sau đó có rất nhiều cố gắng của các nhà toán học để giải bài toán này nhưng đều không đi đến kết quả cuối cùng. Cho đến năm 1976, một nhóm các nhà toán học (K. Appel, W. Haken, J.Koch) đã xây dựng một lời giải dựa trên kết quả do máy tính IBM cung cấp đã khẳng định được giả thiết 4 màu là đúng.

❖ **LIÊN QUAN GIỮA BÀI TOÁN 4 MÀU & SẮC TỐ ĐỒ THỊ PHẲNG.**

Cho một đồ thị phẳng G liên thông, không có đỉnh cô lập. Ta xây dựng một đồ thị đối ngẫu của nó gọi là G^* như sau :

- Mỗi đỉnh x^* của G^* tương ứng đúng với một mặt s của G.
- Mỗi cạnh u^* của G^* nối 2 đỉnh của G^* tương ứng với 2 vùng kề nhau và cắt cạnh chung của hai vùng đó.

G^* được xây dựng như trên là một đồ thị phẳng, và cũng không có đỉnh cô lập.

Chú ý : Đối ngẫu của G^* là G.

❖ **HỆ QUẢ.**

Trong tất cả các bản đồ địa dư, có ít nhất một mặt có đường biên có số cạnh ≤ 5.

Chứng minh.

Chuyển bản đồ địa dư thành đồ thị đối ngẫu. Giả thiết trở thành « có it nhất một đỉnh có bậc $5 \leq$ ». áp dụng Hệ quả 4.2.3. suy ra kết luận của hệ quả trên.

❖ **ĐỊNH LÝ 4 MÀU.**

Mọi đồ thị phẳng có sắc tố $\gamma(G) \leq 4$.