CHƯƠNG 4

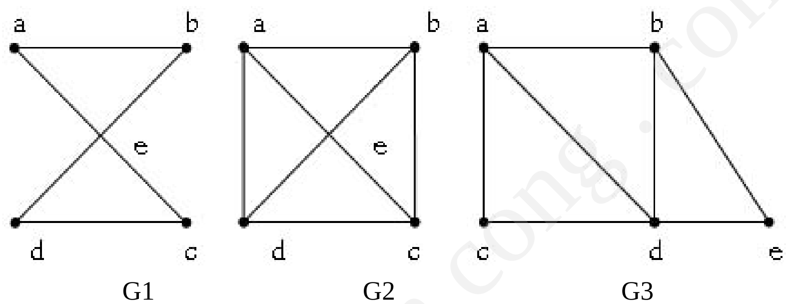
ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON

I. Đồ thị EULER

1. Khái niệm

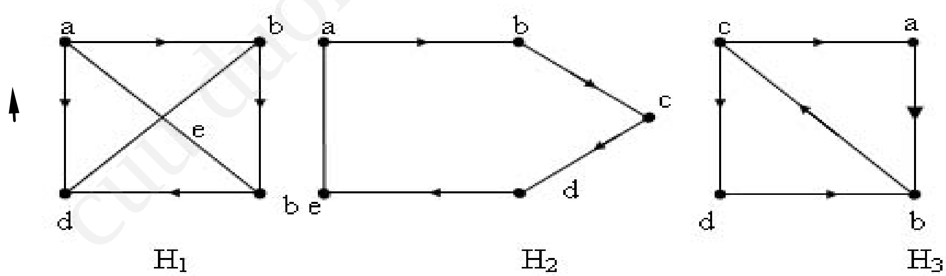
- Đường đi qua mỗi cạnh của đồ thị đúng một lần được gọi là đường đi Euler.
- Chu trình qua mỗi cạnh của đồ thị đúng một lần được gọi là chu trình Euler.
- Đồ thị được gọi là đồ thị Euler nếu nó có chu trình Euler, và gọi là đồ thị nửa Euler nếu nó có đường đi Euler.
- Mọi đồ thị Euler luôn là nửa Euler, nhưng điều ngược lại không luôn đúng.

Ví dụ:
Đồ thị G_1 trong hình 1 là đồ thị Euler vì nó có chu trình Euler a, e, c, d, e, b, a. Đồ thị G_3 không có chu trình Euler nhưng nó có đường đi Euler a, c, d, e, b, d, a, b, vì thế G_3 là đồ thị nửa Euler. Đồ thị G_2 không có chu trình cũng như đường đi Euler.



Hình 1. Đồ thị G_1, G_2, G_3

Ví dụ:
Đồ thị H_2 trong hình 2 là đồ thị Euler vì nó có chu trình Euler a, b, c, d, e, a. Đồ thị H_3 không có chu trình Euler nhưng nó có đường đi Euler c, a, b, c, d, b vì thế H_3 là đồ thị nửa Euler. Đồ thị H_1 không có chu trình cũng như đường đi Euler.



Hình 2. Đồ thị H_1, H_2, H_3

2. Định lý 1 (Định lý Euler):

G là đồ thị vô hướng liên thông.
G là đồ thị Euler ⇔ mọi đỉnh của G đều có bậc chẵn.

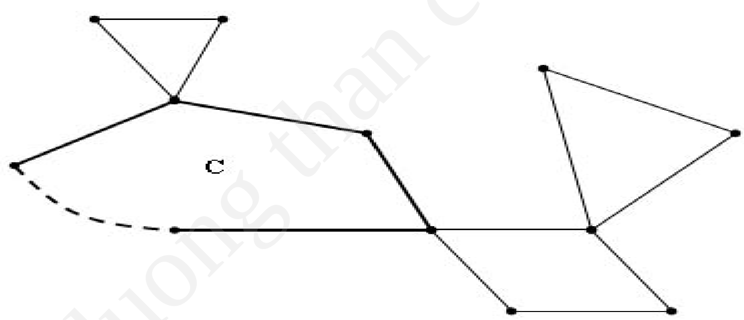
Bổ đề: Nếu bậc của mỗi đỉnh của đồ thị G không nhỏ hơn 2 thì G chứa chu trình.
Cm bổ đề:
Nếu G có cạnh lặp thì khẳng định của bổ đề là hiển nhiên. Vì vậy giả sử G là đơn đồ thị. Gọi v là một đỉnh nào đó của G. Ta sẽ xây dựng đường đi như sau: v -> v_1 -> v_2 -> . . .

trong đó v_1 là đỉnh kề với v, còn với i≥1 chọn v_{i+1} kề với v_i và $v_{i+1} \neq v_{i-1}$ (có thể chọn v_{i+1} như vậy là vì deg(vi) ≥2). Do tập đỉnh của G là hữu hạn , nên sau một số hữu hạn bước ta phải quay lại một đỉnh đã xuất hiện trước đó và là 1 chu trình cần tìm.

Chứng minh định lý:
($\Rightarrow$) Giả sử G là đồ thị Euler tức là tồn tại chu trình Euler P trong G. Mỗi lần chu trình P đi qua một đỉnh nào đó của G thì bậc của đỉnh đó tăng lên 2. Mặt khác mỗi cạnh của đồ thị xuất hiện trong P đúng một lần, suy ra mỗi đỉnh của đồ thị đều có bậc chẵn.

($\Leftarrow$) Quy nạp theo số cạnh của G. Do G liên thông và bậc của mọi đỉnh là số chẵn nên bậc của mỗi đỉnh của nó không nhỏ hơn 2. Từ đó theo bổ đề G phải chứa chu trình C. Nếu C đi qua tất cả các cạnh của G thì nó chính là chu trình Euler. Giả sử C không đi qua tất cả các cạnh của G. Khi đó loại bỏ khỏi G tất cả các cạnh thuộc C ta thu được một đồ thị mới H vẫn có tất cả các đỉnh bậc chẵn. Theo giả thiết qui nạp, trong mỗi thành phần liên thông của H đều tìm được chu trình Euler. Do G là liên thông nên trong mỗi thành phần của H có ít nhất một đỉnh chung với chu trình C.

Vì vậy, ta có thể xây dựng chu trình Euler trong G như sau: bắt đầu từ một đỉnh nào đó của chu trình C, đi theo các cạnh của C cho đến khi gặp đỉnh chung với H, ta sẽ đi theo chu trình Euler của thành phần liên thông của H chứa đỉnh đó. Sau đó lại tiếp tục đi theo cạnh của C cho đến khi gặp phải đỉnh chung với H thì lại theo chu trình Euler của thành phần liên thông tương ứng trong H v.v... (xem hình 3). Quá trình sẽ kết thúc khi ta trở về đỉnh xuất phát , tức là thu được chu trình đi qua mỗi cạnh của đồ thị đúng một lần.



Hình 3. Minh hoạ cho chứng minh định lý 1.

3. Hệ quả :

Cho đồ thị vô hướng liên thông G. G là nửa Euler ⇔ G có không quá 2 đỉnh bậc lẻ.

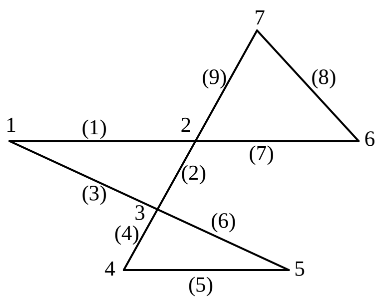
Chứng minh.
($\Leftarrow$) Nếu G có không quá 2 đỉnh bậc lẻ thì số đỉnh bậc lẻ chỉ có thể là 0 hoặc 2 (do số đỉnh bậc lẻ là số chẵn). Nếu G không có đỉnh bậc lẻ thì theo định lý 1, nó là đồ thị Euler. Giả sử G có 2 đỉnh bậc lẻ là u và v. Gọi H là đồ thị thu được từ G bằng cách thêm vào G một đỉnh mới w và hai cạnh (w,u) và (w,v). Khi đó tất cả các đỉnh của H đều có bậc chẵn, vì thế theo định lý 1, nó có chu trình Euler. Xoá bỏ khỏi chu trình này đỉnh w và hai cạnh kề nó ta thu được đường đi Euler trong đồ thị G.

4. Thuật toán tìm chu trình Euler hoặc đường đi Euler

```
void Euler_Cycle(){
        stack=∅ ; CE=∅ ;// CE la tap chua cac dinh theo thu tu cua chu trinh Euler
```

cat dinh xp vao stack
Trong khi stack còn khác rỗng{
  Goi x la phan tu dinh stack ;
  Nếu (x còn đỉnh kề)
  { Chọn một đỉnh y kề x, cất y vào stack;
   Loại bỏ cạnh (x,y) khỏi đồ thị
  }
  Ngược lại //x khong con dinh ke
  { Lấy x ra khỏi stack ; cất x vào CE
  }
}
In CE theo thứ tự ngược
}

Ví dụ : Chọn 1 la dinh xuat phat



							3												
						5	5	5						2					
			1		4	4	4	4	4				7	7	7				
		3	3	3	3	3	3	3	3	3		6	6	6	6	6			
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

Thứ tự các cạnh bị huỷ được ghi trong ngoặc.
CE = {1, 3, 5, 4, 3, 2, 7, 6, 2, 1}: chu trình Euler

5. Thuật toán Flor

Giả sử G là đồ thị Euler, thuật toán đơn giản sau đây cho phép xác định chu trình Euler khi làm bằng tay : Xuất phát từ một đỉnh u nào đó của G ta đi theo các cạnh của nó một cách tuỳ ý chỉ cần tuân thủ 2 qui tắc sau:

- Xoá bỏ cạnh đã đi qua đồng thời xoá bỏ cả những đỉnh cô lập tạo thành.
- Ở mỗi bước ta chỉ đi qua cầu khi không còn cách lựa chon nào khác.

6. Định lý 2 :

G có hướng liên thông mạnh
G là đồ thị Euler ⇔ $Deg^+(v) = deg^-(v)$, $\forall v \in V$.

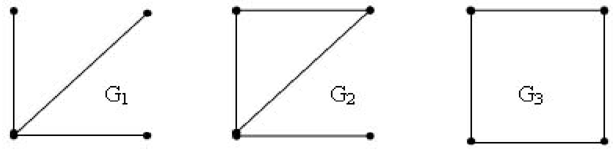
II. Đồ thị HAMILTON

1. Khái niệm

- Đường đi qua tất cả các đỉnh của đồ thị mỗi đỉnh đúng một lần được gọi là đường đi Hamilton.
- Chu trình bắt đầu từ một đỉnh v nào đó qua tất cả các đỉnh còn lại mỗi đỉnh đúng một lần rồi quay trở về v được gọi là chu trình Hamilton.
- Đồ thị G được gọi là đồ thị Hamilton nếu nó chứa chu trình Hamilton và gọi là đồ thị nửa Hamilton nếu nó có đường đi Hamilton.
- Đồ thị Hamilton là nửa Hamilton, nhưng điều ngược lại không đúng.

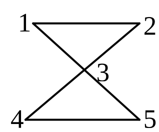
Ví dụ:

G_3 là Hamilton, G_2 là nửa Hamilton , G_1 không là nửa Hamilton.

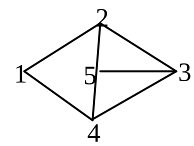


Hình 4. Đồ thị Hamilton G_3, nửa Hamilton G_2 , và G_1 không là nửa Hamilton.

Lưu ý: đồ thị Euler có thể không Hamilton và hamilton có thể không Euler



Euler kg Hamilton



Hamilton kg Euler

2. Định lý 3

> G là đơn đồ thị vô hướng có n đỉnh (n>2).
> $\forall$ đỉnh u, deg(u) $\geq$ n/2 $\Rightarrow$ G là đồ thị Hamilton

3. Định lý 4

> G là đồ thị có hướng liên thông với n đỉnh.
> $\forall$ đỉnh u, $deg^{+}(u) \geq n/2$, $deg^{-}(u) \geq n/2$ $\Rightarrow$ G đồ thị là Hamilton.

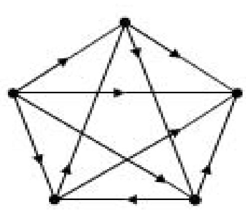
Đồ thị đấu loại là đồ thị có hướng mà hai đỉnh bất kỳ được nối với nhau bởi đúng một cung. Đồ thị đấu loại có thể dùng để biểu diễn kết quả thi đấu một trò chơi mà không cho phép hoà.

4. Định lý 5

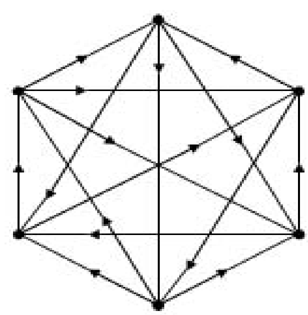
> i) Mọi đồ thị đấu loại là nửa Hamilton.
> ii) Mọi đồ thị đấu loại liên thông mạnh là Hamilton.

Ví du:

Đồ thị đấu loại D_5, D_6 được cho trong hình 5.



D_5



D_6

Hình 5. Đồ thị đấu loại D_5, đấu loại liên thông mạnh D_6

5. Thuật toán liệt kê tất cả các chu trình/đường đi Hamilton

```
void Try(int i){
	for(int j=1;j<=n;j++){
		if(tham[j]==0&&a[x[i-1]][j]==1){
			x[i]=j;
			tham[j]=1;
			if(i==n){
				if(a[x[n]][x[1]]==1){
					cout<<"CT: ";
					for(int k=1;k<=n;k++)  cout<<x[k]<<"->";
					cout<<x[1]<<endl;
				}
				else{
					cout<<"DD: ";
					for(int k=1;k<n;k++) cout<<x[k]<<"->";
					cout<<x[n]<<endl;
				}
			}else	Try(i+1,a,x,tham,n);
			tham[j]=0;
		}
	}
}
void Hamilton(int dxp){
	for(int i=1;i<=n;i++)	tham[i]=0;
	x[1]=dxp;
	tham[dxp]=1;
	Try(2);
}
```

- **Bài tập chương 4**

Bài 1:

Kiểm tra đồ thị vô hướng có là Euler hoặc nửa Euler hay không? Nếu có hãy in ra đường đi hoặc chu trình Euler.

Bài 2:

In ra tất cả các đường đi hoặc chu trình Hamilton nếu có của một đồ thị cho trước.

Bài 3:

Sử dụng thuật toán tìm chu trình Hamilton, giải bài toán "Người du lịch", in ra lộ trình tối ưu.

Biết rằng bảng chi phí lưu trong file văn bản dạng sau:

```
4				//4 thanh pho
0	4	1	3
1	0	-1	2
-1	1	0	4
2	3	5	1
```

(a[i][j]=-1 là không có đường đi trực tiếp từ i đến j.)