CHƯƠNG 3. ĐỒ THỊ EULER VÀ HAMILTON

3.1 Chu trình và đường đi Euler

1) Định nghĩa: Cho đồ thị G= (V, E).

- Chu trình đơn chứa tất cả các cạnh của G được gọi là *chu trình Euler* ⇒ G là đồ thị Euler.

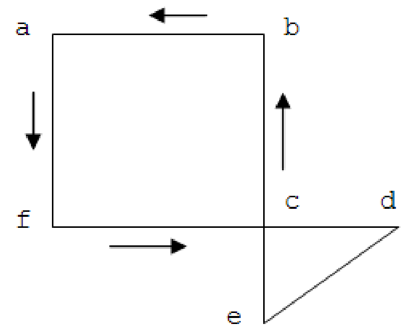
- *Đường đi Euler* trong G là đường đi đơn chứa mọi cạnh của G ⇒ G là đồ thị nửa Euler.

2) Điều kiện:

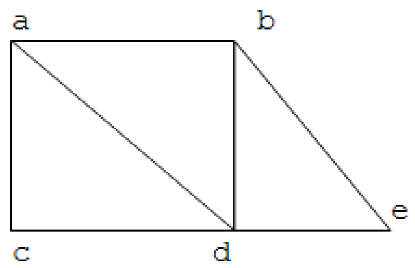
- Đồ thị liên thông vô hướng G là đồ thị Euler ⇔ mọi đỉnh v ∈ V có bậc chẵn.
- Đồ thị liên thông vô hướng G là đồ thị nửa Euler ⇔ số đỉnh v ∈ V có bậc lẻ không vượt quá 2.
- Đồ thị có hướng, liên thông yếu G là đồ thị Euler ⇔ mọi đỉnh v ∈ V có bậc-vào và bậc-ra bằng nhau.
- Đồ thị có hướng, liên thông yếu G là đồ thị nửa Euler ⇔ số đỉnh v ∈ V có bậc-vào và bậc-ra chênh lệch

nhau 1 đon vị không vượt quá 2.

3) Ví dụ:



Đồ thị có chu trình Euler:
a-f-c-d-e-c-b-a



Đồ thị có đường đi Euler:
a-b-a-c-d-e-b

4) Thuật toán tìm chu trình/ đường đi Euler

Input: Cho đồ thị G = (V,E) gồm n đỉnh biểu diễn bởi ma trận kề.

Output: Hãy tìm chu trình/đường đi Euler của đồ thị G nếu có.

(1) Kiểm tra G có thỏa mãn điều kiện hay không? Nếu G không thỏa mãn điều kiện thì kt= 0, nếu có chu trình Euler thì kt= 1; nếu có đường đi Euler thì kt= 2.

(2) Nếu kt= 0 $\Rightarrow$ thông báo đồ thị không có chu trình/đường đi Euler và dừng;

Nếu kt= 1 $\Rightarrow$ chọn u là đỉnh cho trước và chuyển sang (3);

Nếu kt= 2 $\Rightarrow$ u là đỉnh có hiệu bán bậc ra và bán bậc vào bằng 1 (đỉnh bậc lẻ) ; chuyển sang 3 ;

(3) Xây dựng chu trình/đường đi Euler trong G:

(3.1) Tạo mảng CE để ghi chu trình Euler và Stack để xếp các đỉnh sẽ xét. Xếp đỉnh u vào Stack;

(3.2) Xét đỉnh v nằm trên cùng của Stack và thực hiện:

- Nếu v là đỉnh cô lập thì lấy v ra khỏi Stack và đưa vào CE.
- Nếu v có đỉnh kề là x thì đưa x vào Stack sau đó xóa cạnh nối v với x;

(3.3) Quay lại (3.2) cho tới khi stack rỗng;

(4) Xuất chu trình/đường đi Euler chứa trong CE theo thứ tự ngược lại.

Cài đặt chương trình tìm chu trình Euler với G vô hướng:

int lt(int a[][], int n)
 {int x;
  x = tplt();
  if (x > 1) return 0;
  else return 1;
  }
	int bc(int a[][], int n)
	{int i, j, deg;
	 for(i= 1; i<= n; i++)
	  {deg=0;
	   for(j= 1; j<= n; j++)
	    deg+=a[i][j];
	    if (deg % 2 > 0) return (0);
	  }
	 return (1); }

```
void ceu(int a[][], int n)
 {int st[100*100], i, j, k, h, t;
  t= 1; st[t]= 1; k= 0;
 while (t> 0)
  {h= st[t]; j= 0;
   for (i= 1; i<= n; i++)
      if (a[h][i] ==1)
        {t++; st[t]= i; j= i;
          a[h][i]= 0; a[i][h]= 0; break}
       if (j== 0)
        {k++, ce[k]= h; t--;}
   }
  for (i= 1; i<= k; i++)
   cout << ce[i] << " ";
 }
```

```
void main()
{clrscr();
if (  lt(a, n)== 0 || bc(a, n) == 0)
    cout << "KHONG XET\n";
else ceu(a, n);
 getch() ; }
```

BÀI TẬP

1. Cho đơn đồ thị vô hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng ma trận kề hình bên.

Hãy thực hiện:

a) Chứng minh đồ thị G đã cho là đồ thị nửa Euler?

b) Tìm đường đi Euler của G?

	1	2	3	4	5	6		8	9	
1	0	1	0	0	1	0	0	1	0	1
2	1	0	1	1	0	1	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0
4	0	1	1	0	0	1	0	0	0	0
5	1	0	0	0	0	1	1	1	1	1
6	0	1	0	1	1	0	1	0	0	0
7	0	0	0	0	1	1	0	1	1	0
8	1	0	0	0	1	0	1	0	0	0
9	0	0	0	0	1	0	1	0	0	0
0	1	0	0	0	1	0	0	0	0	0

2. Cho đơn đồ thị có hướng G = <V, E> gồm 10 đỉnh được biểu diễn dưới dạng ma trận kề như hình bên.

Hãy thực hiện:

a) Chứng minh đồ thị G đã cho là đồ thị Euler?

b) Tìm một chu trình Euler của G bắt đầu từ đỉnh 1?

	1	2	3	4	5	6	7	8	9	0
1	0	1	1	0	0	0	0	0	0	0
2	0	0	1	1	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	1	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	1	0	0
7	0	0	0	1	0	0	0	1	0	0
8	1	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0

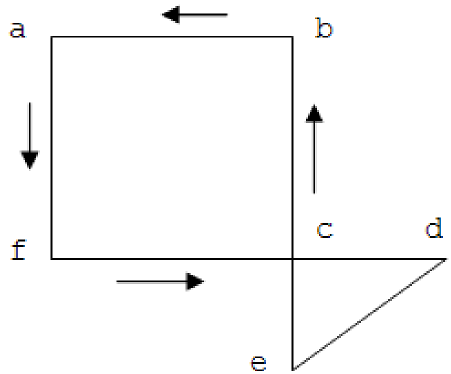
3.2 Chu trình và đường đi Hamilton

1) Định nghĩa: Cho đồ thị G = (V, E).

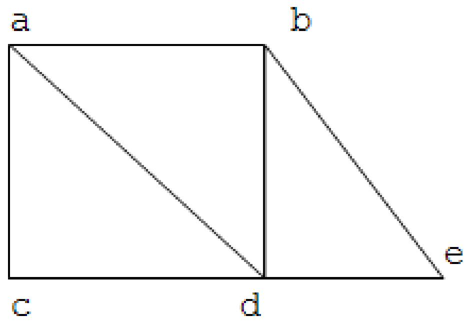
- Chu trình đơn đi qua tất cả các đỉnh của G, mỗi đỉnh 1 lần gọi là *chu trình Hamilton* ⇒ G là đồ thị Hamilton.

- *Đường đi Hamilton* trong G là đường đi đi qua tất cả các đỉnh của G, mỗi đỉnh 1 lần ⇒ G là đồ thị nửa Hamilton.

2) Ví dụ:



Đồ thị có đường đi Hamilton:
d-e-c-b-a-f



Đồ thị có chu trình Hamilton:
a-b-e-d-c-a

3) Liệt kê tất cả chu trình Hamilton trong đồ thị vô hướng

Input: $G = (V, E)$, $V = \{v_1, v_2, ..., v_n\}$.

Output: Dãy đỉnh $x_0, x_1, ..., x_n$ với $x_n = x_0$.

Giải thuật đệ qui quay lui :

Khởi tạo: mảng vs[i] = 0 để đánh dấu những đỉnh đã xét. Chọn $u \in V$ bất kỳ làm đỉnh xuất phát và đặt $x_0 = u$; k= 0;

Lặp quay lui:

- Trong các đỉnh v_i kề x_{k-1} có $vs[v_i] = 0$, chọn đỉnh v_h có chỉ số nhỏ nhất và đặt $x_k = v_h$; $vs[v_h] = 1$;
- Tại bước k nào đó không chọn được đỉnh kề $\Rightarrow$ quay lại bước k-1, bỏ đánh dấu đỉnh đã chọn tại bước k-1 và chọn đỉnh khác tiếp theo nếu có thể, nếu chọn được thì chuyển sang bước k+1,

nếu không chọn được thì quay về bước k-1,...

- Nếu k= n và chọn được x_n $\Rightarrow$ nếu $x_n = x_0$ thì xuất một chu trình Hamilton tìm được.