TOÁN RỜI RẠC 2

CHƯƠNG 3

Giảng viên: Vũ Văn Thỏa

CHƯƠNG 3
ĐỒ THỊ EULER VÀ HAMILTON

- **Chu trình và đường đi Euler**
- **Chu trình và đường đi Hamilton**

3.1 Chu trình và đường đi Euler

- Định nghĩa
- Chu trình và đường đi Euler của đồ thị vô hướng
- Chu trình và đường đi Euler của đồ thị có hướng

3.1.1 Định nghĩa

Cho đồ thị G= (V, E).

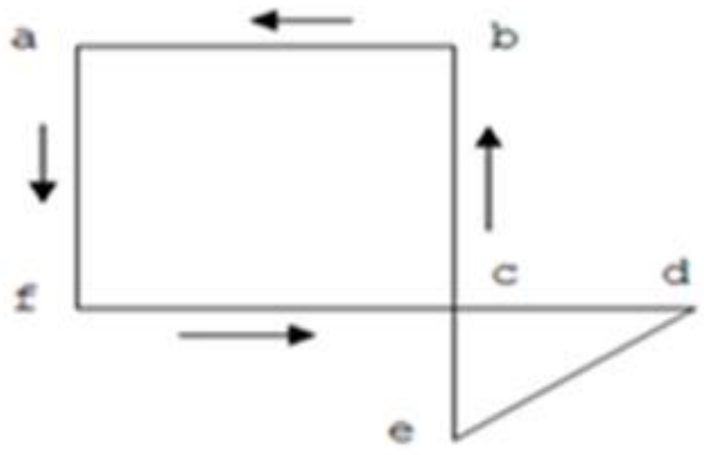
- Chu trình đơn chứa tất cả các cạnh của G được gọi là ***chu trình Euler*.**

G là ***đồ thị Euler*** ⇔ G chứa chu trình Euler.

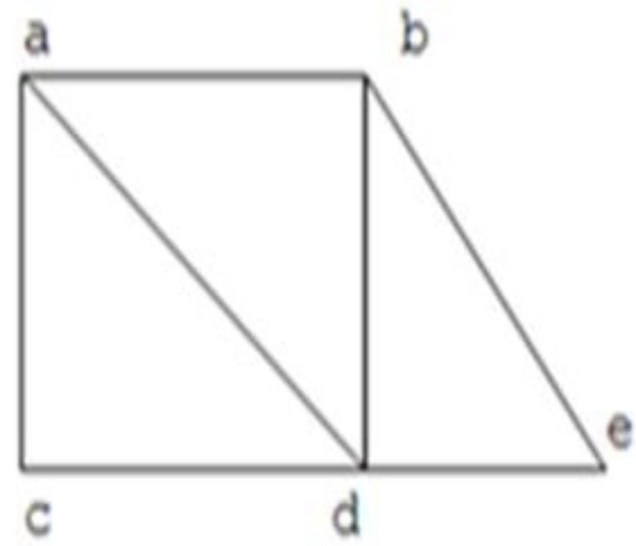
- *Đường đi Euler* trong G là đường đi đơn chứa tất cả các cạnh của G.

G là ***đồ thị nửa Euler*** ⇔ G chứa đường đi Euler.

Ví dụ 1: Đồ thị Euler và nửa Euler vô hướng



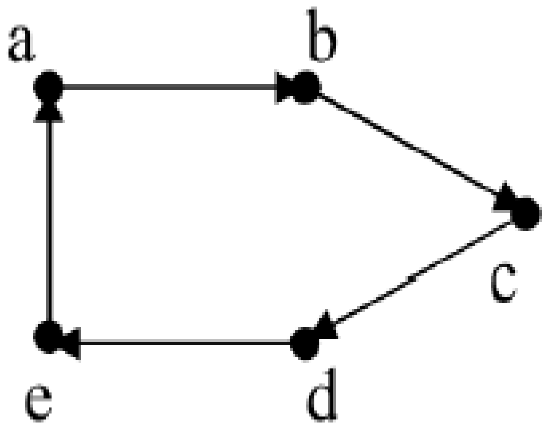
Đồ thị vô hướng có chu trình Euler:
a-f-c-d-e-c-b-a



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

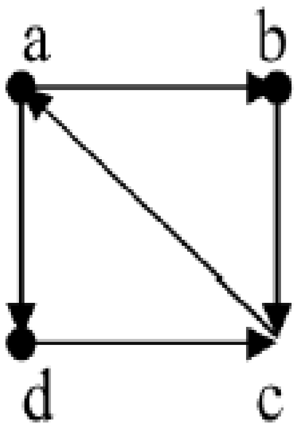
có 2 đỉnh có bậc lẻ

Ví dụ 2: Đồ thị Euler và nửa Euler có hướng



Đồ thị Euler có hướng

a→b→c→d→e→a



Đồ thị nửa Euler có hướng

a→b→c→a→d→c

3.1.2 Chu trình và đường đi Euler của đồ thị vô hướng

- Điều kiện tồn tại
- Thuật toán tìm chu trình và đường đi Euler

Điều kiện tồn tại

- **Định lý 1**

- Đồ thị vô hướng G là đồ thị Euler ⇔ G *liên thông* và mọi đỉnh $v \in V$ *có bậc chẵn*.

- Đồ thị vô hướng G là đồ thị nửa Euler ⇔ G *liên thông* và số đỉnh $v \in V$ có *bậc lẻ không vượt quá 2*.

Ví dụ 3: Chứng minh G vô hướng gồm 13 đỉnh biểu diễn bởi ma trận kề là Euler

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	0	0	1	1	0	0	1	0	0
5	0	1	1	0	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	0	0	0	0	1	1	0	1	1	0
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	0	0	1	0

Bfs(1)= {1(0); 2(1), 6(1); 3(2), 5(2); 7(6); 4(3), 11(3); 8(7); 10(11), 12(11); 9(8); 13(12)} = V $\Rightarrow$ G là đồ thị liên thông.

Tính bậc

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	0	0	1	1	0	0	1	0	0
5	0	1	1	0	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	0	0	0	0	1	1	0	1	1	0
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	0	0	1	0

Tính bậc của các đỉnh:

deg(1)= 2; deg(2)= 4;

deg(3)= 4; deg(4)= 4;

deg(5)= 4; deg(6)= 4;

deg(7)= 4; deg(8)= 4;

deg(9)=4; deg(10)= 4;

deg(11)= 4; deg(12)= 4; deg(13)= 2

$\Rightarrow$ Tất cả các đỉnh của G đều có bậc chẵn.

Kết luận: G là đồ thị Euler.

<u>Ví dụ 4</u>: Chứng minh G vô hướng gồm 13 đỉnh biểu diễn bởi ma trận kề là nửa Euler

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	1	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	1	0	1	1	0	1	1	0	0
5	1	1	1	1	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	1	0	0	0	1	1	0	1	1	1
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	1	0	1	0

Dfs(1)= {1(0); 2(1); 3(2); 4(3); 5(4); 6(5); 7(6); 8(7); 9(8); 10(9), 11(10); 12(11); 13(12)} = V $\Rightarrow$ G là đồ thị liên thông.

Tính bậc của các đỉnh

deg(1)= 3; deg(2)= 4;
deg(3)= 4; deg(4)= 6;
deg(5)= 6; deg(6)= 4;
deg(7)= 4;deg(8)= 4;
deg(9)=4; deg(10)= 6;
deg(11)= 4;deg(12)= 4;
deg(13)= 3

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	1	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	1	0	1	1	0	1	1	0	0
5	1	1	1	1	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	1	0	0	0	1	1	0	1	1	1
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	1	0	1	0

$\Rightarrow$ Có 2 đỉnh bậc lẻ là 1 và 13; Tất cả các đỉnh còn lại của G đều có bậc chẵn.

Kết luận: G là đồ thị nửa Euler.

Thuật toán tìm chu trình/đường đi Euler

- **Input**: Cho đồ thị vô hướng G = (V,E) gồm n và m cạnh;
- **Output**: Chu trình/đường đi Euler của đồ thị G nếu có;

Mô tả thuật toán

- **Thuật toán:** ***Tìm chu trình/đường đi Euler***

Bước 1(Kiểm tra điều kiện): 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;

deg lẻ = 2.

Bước 2:

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

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

(2.3) Nếu kt= 2 $\Rightarrow$ chọn u là đỉnh bậc lẻ có số hiệu nhỏ nhất và chuyển sang bước 3;

Mô tả thuật toán

Bước 3 (Xây dựng chu trình/đường đi Euler bắt đầu từ đỉnh u):

(3.1) Tạo mảng CE để ghi chu trình/ đường đi 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;

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

Ví dụ 5: Tìm chu trình Euler từ u= 1 của G vô hướng 13 đỉnh biểu diễn bởi ma trận kề

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	0	0	1	1	0	0	1	0	0
5	0	1	1	0	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	0	0	0	0	1	1	0	1	1	0
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	0	0	1	0

Số đỉnh của đồ thị n= 13, số cạnh m= 24

Tìm chu trình Euler bắt đầu tại u= 1:

Lập bảng

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	0	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	0	0	1	1	0	0	1	0	0
5	0	1	1	0	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	0	0	0	0	1	1	0	1	1	0
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	0	0	1	0

Bước	Stack	Cạnh được duyệt	CE
1	1	∅	∅
2	1,2,3,4,7,5,2,6,1	(1,2);(2,3);(3,4);(4,7);(7,5);(5,2);(2,6);(6,1)	
3			1
4	1,2,3,4,7,5,2,6,5,3,11,4,8,7,6	(6,5);(5,3);(3,11);(11,4);(4,8);(8,7);(7,6)	
5			1,6,7
6	1,2,3,4,7,5,2,6,5,3,11,4,8,9,10,8	(8,9);(9,10);(10,8)	
7	1,2,3,4,7,5,2,6,5,3,11,4,8,9,10,11,12, 9,13,12,10	(10,11);(11,12),(12,9);(9,13);(13,12); (12,10)	1,6,7,8
8	∅		1,6,7,8,10,12,13,9,12,11,10,9,8, 4,11,3,5,6,2,5,7,4,3,2,1

Kết luận: Chu trình Euler bắt đầu tại u= 1 là 1- 2- 3- 4- 7- 5- 2- 6- 5- 3- 11- 4- 8- 9- 10- 11- 12- 9- 13- 12- 10- 8- 7- 6- 1

Ví dụ 6: Tìm đường đi Euler từ u= 1 của G vô hướng 13 đỉnh biểu diễn bởi ma trận kề

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	1	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	1	0	1	1	0	1	1	0	0
5	1	1	1	1	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	1	0	0	0	1	1	0	1	1	1
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	1	0	1	0

Số đỉnh của đồ thị n= 13, số cạnh m= 28

Tìm đường đi Euler bắt đầu tại đỉnh bậc lẻ nhỏ nhất u= 1:

- Lập bảng:

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	1	1	0	0	0	0	0	0	0
2	1	0	1	0	1	1	0	0	0	0	0	0	0
3	0	1	0	1	1	0	0	0	0	0	1	0	0
4	0	0	1	0	1	0	1	1	0	1	1	0	0
5	1	1	1	1	0	1	1	0	0	0	0	0	0
6	1	1	0	0	1	0	1	0	0	0	0	0	0
7	0	0	0	1	1	1	0	1	0	0	0	0	0
8	0	0	0	1	0	0	1	0	1	1	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	1	1
10	0	0	0	1	0	0	0	1	1	0	1	1	1
11	0	0	1	1	0	0	0	0	0	1	0	1	0
12	0	0	0	0	0	0	0	0	1	1	1	0	1
13	0	0	0	0	0	0	0	0	1	1	0	1	0

Bước	Stack	Cạnh đã duyệt	CE
1	1	Ø	Ø
2	1, 2, 3, 4, 5, 1, 6, 2, 5, 3, 11, 4, 7, 5, 6, 7, 8, 4, 10, 8, 9, 10, 11, 12, 9, 13, 10, 12, 13	(1,2); (2,3); (3,4); (4,5); (5,1); (1.6); (6,2); (2,5); (5,3); (3,11); (11,4); (4,7); (7,5); (5,6); (6,7); (7,8); (8,4); (4,10); (10,8); (8,9); (9,10); (10,11); (11,12); (12,9); (9,13); (13,10); 10,12); (12,13)	
3	Ø		13, 12, 10, 13, 9, 12, 11, 10, 9, 8, 10, 4, 8, 7, 6, 5, 7, 4, 11, 3, 5, 2, 6, 1, 5, 4, 3, 2, 1

Kết luận: Đường đi Euler bắt đầu tại u=1 là 1- 2- 3- 4- 5- 1- 6- 2- 5- 3- 11- 4- 7- 5- 6- 7- 8- 4- 10- 8- 9- 10- 11- 12- 9- 13- 10- 12- 13

Cài đặt thuật toán

- Hàm kiểm tra điều kiện của G vô hướng

```
int kt(int a[][], int n);
```

- Hàm xây dựng chu trình/đường đi Euler

```
void ceu(int a[][], int n, int u);
```

Hàm kiểm tra điều kiện của G vô hướng

```
int a[100][100], n, u, vs[100], e[100]. s[10000], ce[10000];
int kt(int a[][], int n) {int v;
   for (v= 1; v<= n; v++) { vs[v]= 0; e[v]= 0;}
   DfsDequy(1);    int ok= 1;
   for (v= 1; v<= n; v++) if (vs[v] == 0){ ok= 0; break;}
      if (ok == 0) return(0); int bl= 0;
       for (v= 1; v<= n; v++) {int deg= 0;
           for (int i= 1; i<= n; i++)  if (a[v][i] == 1) deg++;
           if (deg%2 == 1) { bl++;
               if (bl> 2) return(0); if (bl == 1){u= v; ok= 2;}
       return(ok);
}
```

Hàm xây dựng chu trình/đường đi Euler

```
void ceu(int a[][], int n, int u) {int top=0, v;
   top++; s[top]= u; k= 0;
  while (top > 0)  {int v= s[top];
       int ok= 1;
     for (int x= 1; x<= n; x++)
        if (a[v][x] ==1){top++; s[top]= x; ok= 0;
            a[v][x]= 0; a[x][v]= 0; break;
                  }
         if (ok== 1) {k++, ce[k]= v; top--;
                }
          }
     for (v= k; v> 0; v--)  cout << ce[v] << " ";
}
```

3.1.3 Chu trình và đường đi Euler của đồ thị có hướng

- Điều kiện tồn tại
- Thuật toán tìm chu trình và đường đi Euler

Điều kiện tồn tại

- **Định lý 2**

- Đồ thị có hướng G là đồ thị Euler ⇔ G *liên thông yếu* và mọi đỉnh $v \in V$ có *deg-(v) = deg+(v)*.

- Đồ thị có hướng G là đồ thị nửa Euler ⇔ G *liên thông yếu* và số đỉnh $v \in V$ *có bán bậc-vào và bán bậc-ra chênh lệch nhau 1 đơn vị không vượt quá 2*.

Ví dụ 7: Chứng minh G có hướng gồm 10 đỉnh biểu diễn bởi danh sách kề là Euler

Ke(1) = {4}	Ke (6) = {2}
Ke(2) = {3, 5}	Ke (7) = {1}
Ke(3) = {4, 8}	Ke (8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke (5) = {6}	Ke(10) = {3, 7}

Xét đồ thị vô hướng nền của G.

Bfs(1)= {1(0); 4(1), 7(1); 2(4), 3(4), 10(4); 5(2), 6(2); 8(3); 9(10)} = V $\Rightarrow$ G là đồ thị liên thông yếu.

Tính bán bậc của các đỉnh

Ke(1) = {4}	Ke (6) = {2}
Ke(2) = {3, 5}	Ke (7) = {1}
Ke(3) = {4, 8}	Ke (8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke (5) = {6}	Ke(10) = {3, 7}

deg-(1) = deg+(1) = 1; deg-(2) = deg+(2) = 2; deg-(3) = deg+(3) = 2;
deg-(4) = deg+(4) = 2; deg-(5) = deg+(5) = 1; deg-(6) = deg+(6) = 1;
deg-(7) = deg+(7) = 1; deg-(8) = deg+(8) = 1; deg-(9) = deg+(9) = 1;
deg-(10) = deg+(10) = 2

$\Rightarrow$ Tất cả các đỉnh của G đều có bán bậc vào bằng bán bậc ra.

Kết luận: G có hướng là đồ thị Euler.

Ví dụ 8: Chứng minh G có hướng gồm 10 đỉnh dưới dạng danh sách kề là nửa Euler

Ke(1) = {4}	Ke (6) = {2}
Ke(2) = {3, 5, 7}	Ke (7) = {1}
Ke(3) = {4, 8}	Ke (8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke (5) = {6}	Ke(10) = {3, 7}

Xét đồ thị vô hướng nền của G.

Dfs(1)= {1(0); 4(1); 2(4); 3(2); 8(3); 9(8); 10(9); 7(10); 5(2); 6(5)} = V $\Rightarrow$ G là đồ thị liên thông yếu.

Tính bán bậc của các đỉnh

Ke(1) = {4}	Ke (6) = {2}
Ke(2) = {3, 5, 7}	Ke (7) = {1}
Ke(3) = {4, 8}	Ke (8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke (5) = {6}	Ke(10) = {3, 7}

deg-(1) = deg+(1) = 1; deg-(2) = 2; deg+(2) = 3; deg-(3) = deg+(3) = 2;
deg-(4) = deg+(4) = 2; deg-(5) = deg+(5) = 1; deg-(6) = deg+(6) = 1;

deg-(7) = 2, deg+(7) = 1; deg-(8) = deg+(8) = 1; deg-(9) = deg+(9) = 1;
deg-(10) = deg+(10) = 2

Hai đỉnh 2 và 7 có bán bậc vào và ra chênh lệch nhau 1.

Tất cả các đỉnh còn lại của G đều có bán bậc vào bằng bán bậc ra.

Kết luận: G có hướng là đồ thị nửa Euler.

Thuật toán tìm chu trình/đường đi Euler

- **Input**: Cho đồ thị có hướng G = (V,E) gồm n đỉnh và m cạnh;
- **Output**: Chu trình/đường đi Euler của đồ thị G nếu có;

Mô tả thuật toán

- **Thuật toán:** ***Tìm chu trình/đường đi Euler***

Bước 1(Kiểm tra điều kiện): 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;

Bước 2:

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

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

(2.3) Nếu kt= 2 $\Rightarrow$ chọn u là đỉnh có bán bậc ra nhiều hơn bán bậc vào và chuyển sang bước 3;

đơn vị → kt

nh bc cng có deg(-u) - deg(+u) = 1 còn li

Mô tả thuật toán

Bước 3 (Xây dựng chu trình/đường đi Euler bắt đầu từ đỉnh u):

(3.1) Tạo mảng CE để ghi chu trình/ đường đi 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;

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

Ví dụ 9: Tìm chu trình Euler từ u= 1 của G có hướng gồm 10 đỉnh dưới dạng danh sách kề

Ke(1) = {4}	Ke(6) = {2}
Ke(2) = {3, 5}	Ke(7) = {1}
Ke(3) = {4, 8}	Ke(8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke(5) = {6}	Ke(10) = {3, 7}

Giải

Số đỉnh n= 10

Số cạnh m= 14

Lập bảng:

Ke(1) = {4}	Ke(6) = {2}
Ke(2) = {3, 5}	Ke(7) = {1}
Ke(3) = {4, 8}	Ke(8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke(5) = {6}	Ke(10) = {3, 7}

Bước	Stack	Cạnh đã duyệt	CE
1	1	Ø	Ø
2	1, 4, 2, 3, 4, 10, 3, 8, 9, 10, 7, 1	(1,4); (4,2); (2,3); (3,4); (4,10); (10,3); (3,8); (8,9); (9,10); (10,7); (7,1)	
3			1, 7, 10, 9, 8, 3, 10, 4, 3
4	1, 4, 2, 5, 6, 2	(2,5); (5,6); (6,2)	
5	Ø		1, 7, 10, 9, 8, 3, 10, 4, 3, 2, 6, 5, 2, 4, 1

Kết luận: Chu trình Euler bắt đầu tại u= 1 là 1- 4- 2- 5- 6- 2- 3- 4- 10- 3- 8- 9- 10- 7- 1

Ví dụ 10: Tìm đường đi Euler của G có hướng gồm 10 đỉnh dưới dạng danh sách kề

Ke(1) = {4}	Ke(6) = {2}
Ke(2) = {3, 5, 7}	Ke(7) = {1}
Ke(3) = {4, 8}	Ke(8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke(5) = {6}	Ke(10) = {3, 7}

Số đỉnh n= 10. Số cạnh m= 14

Có deg-(2)= 2, deg+(2)= 3; deg-(7)= 2, deg+(7)= 1 $\Rightarrow$ chọn đỉnh xuất phát của đường đi Euler là u = 2.

Lập bảng:

Ke(1) = {4}	Ke(6) = {2}
Ke(2) = {3, 5, 7}	Ke(7) = {1}
Ke(3) = {4, 8}	Ke(8) = {9}
Ke(4) = {2, 10}	Ke(9) = {10}
Ke(5) = {6}	Ke(10) = {3, 7}

Bước	Stack	Cạnh đã duyệt	CE
1	2	Ø	Ø
2	2, 3, 4, 2, 5, 6, 2, 7, 1, 4, 10, 3, 8, 9, 10, 7	(2,3); (3,4); (4,2); (2,5); (5,6); (6,2); (2,7); (7,1); (1,4); (4,10); (10,3); (3,8); (8,9); (9,10); (10,7)	
3	Ø		7, 10, 9, 8, 3, 10, 4, 1, 7, 2, 6, 5, 2, 4, 3, 2

Kết luận: Đường đi Euler bắt đầu tại u= 2 là 2- 3- 4- 2- 5- 6- 2- 7- 1- 4- 10- 3- 8- 9- 10- 7

Cài đặt thuật toán

- Hàm kiểm tra điều kiện của G có hướng

```
int kt(int a[][], int n);
```

- Hàm xây dựng chu trình/đường đi Euler

```
void ceu(int a[][], int n, int u);
```

Hàm kiểm tra điều kiện của G có hướng

```
int a[100][100], n, u, vs[100], s[10000], ce[10000];
int kt(int a[][], int n) {int v, top= 0, dem= 0;
   top++; s[top]= 1; dem++;
   while (top> 0){ int v= s[top], ok= 1;;
       for (int i= 1; i<= n; i++)  if ((a[v][i] == 1 || a[i][v]== 1) && vs[i]== 0){
          top++; s[top]= i; vs[i] =1; dem++; ok= 0; break;}
               if (ok == 1) top--; }
      if (dem< n) return(0); int x= 0, y= 0;
       for (v= 1; v<= n; v++) {int d1= 0, d2= 0;
           for (int i= 1; i<= n; i++){  if (a[i][v] == 1) d1++; if (a[v][i] == 1) d2++;}
    if (d1!= d2){   if (abs(d1-d2) > 1) return(0); else{ if (x> 0 && y> 0) return(0);
               else { if (d2> d1) x= v; else y= v;}
      if (x== 0 && y== 0) return(1); else{ u= x; return(2); }  }
}
```

Hàm xây dựng chu trình/đường đi Euler

```
void ceu(int a[][], int n, int u) {int top=0, v;
   top++; s[top]= u; k= 0;
  while (top > 0)  {int v= s[top];
       int ok= 1;
     for (int x= 1; x<= n; x++)
        if (a[v][x] ==1){top++; s[top]= x; ok= 1;
            a[v][x]= 0; break;
                  }
        if (ok== 1) {k++, ce[k]= v; top--;
                }
         }
     for (v= k; v> 0; v--)  cout << ce[v] << " ";
}
```

3.2 Chu trình và đường đi Hamilton

- Định nghĩa
- Thuật toán tìm chu trình và đường đi Hamilton

3.2.1 Định nghĩa

Cho đồ thị G = (V, E).

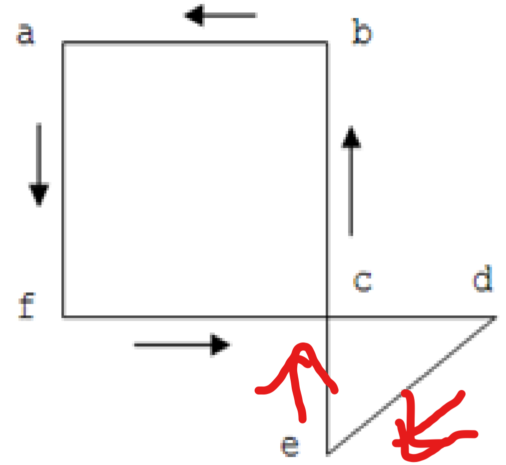
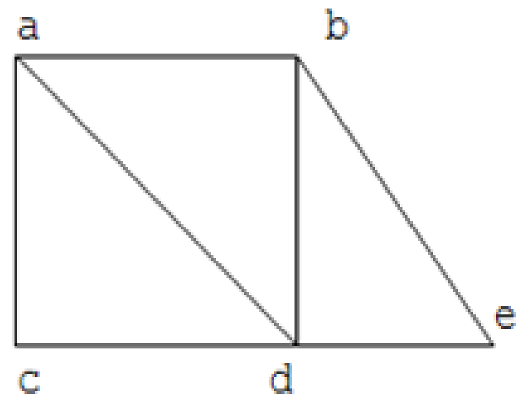
- Chu trình đơn đi qua tất cả các đỉnh của G, mỗi đỉnh 1 lần gọi là *chu trình Hamilton*

G là đồ thị Hamilton ⇔ G chứa chu trình Hamilton

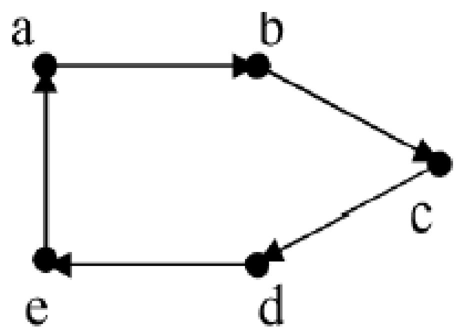
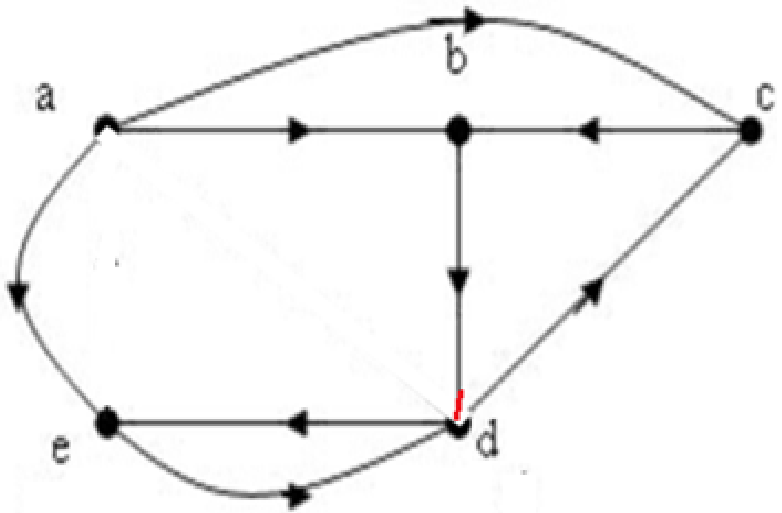
- *Đường đi Hamilton* trong G là đường đi ~~đi~~ qua tất cả các đỉnh của G, mỗi đỉnh 1 lần

G là đồ thị nửa Hamilton ⇔ G chứa đường đi Hamilton

Ví dụ 11: Đồ thị vô hướng Hamilton và nửa Hamilton

 Đồ thị vô hướng có đường đi Hamilton: d-e-c-b-a-f	 Đồ thị vô hướng có chu trình Hamilton: a-b-e-d-c-a

Ví dụ 12: Đồ thị có hướng Hamilton và nửa Hamilton

 Đồ thị Hamilton có hướng a→b→c→d→e→a	 Đồ thị nửa Hamilton có hướng a→c→b→d→e

Nhận xét:

- Cho đến nay, chưa tìm ra được một tiêu chuẩn để nhận biết một đồ thị có phải là đồ thị Hamilton (nửa Hamilton) hay không

- Người ta chỉ có thể chỉ ra một số điều kiện đủ để đồ thị là Hamilton (thường là đồ thị đủ nhiều cạnh).

- Cho đến nay, cũng vẫn chưa có thuật toán hiệu quả để kiểm tra một đồ thị có phải là đồ thị Hamilton hay không

Thuật toán liệt kê các chu trình Hamilton

- **Input**: Cho đồ thị G = (V,E) gồm n đỉnh và m cạnh;

  Một đỉnh u $\in$ G;
- **Output**: Tất cả các chu trình Hamilton (nếu có) bắt đầu tại u của G dưới dạng x[1], …, x[n];

Thuật toán sử dụng đệ quy quay lui

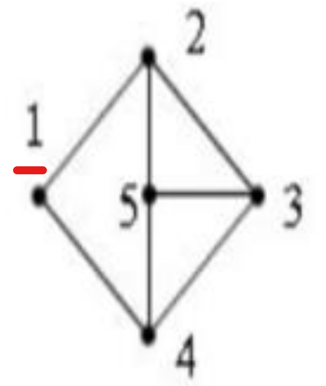
```
Hamilton(k){
    for v∈ Ke(x[k-1]){
      if (k= n+1) { if (v= x[1]) GhiNhan(x[1], …, x[n]);}
        else
          if  (vs[v]= 0) {
              x[k]= v; vs[v]= 1;
              Hamilton(k+1);
               vs[v]= 0; }
              }
      }
```

Thuật toán liệt kê các chu trình Hamilton bắt đầu tại u

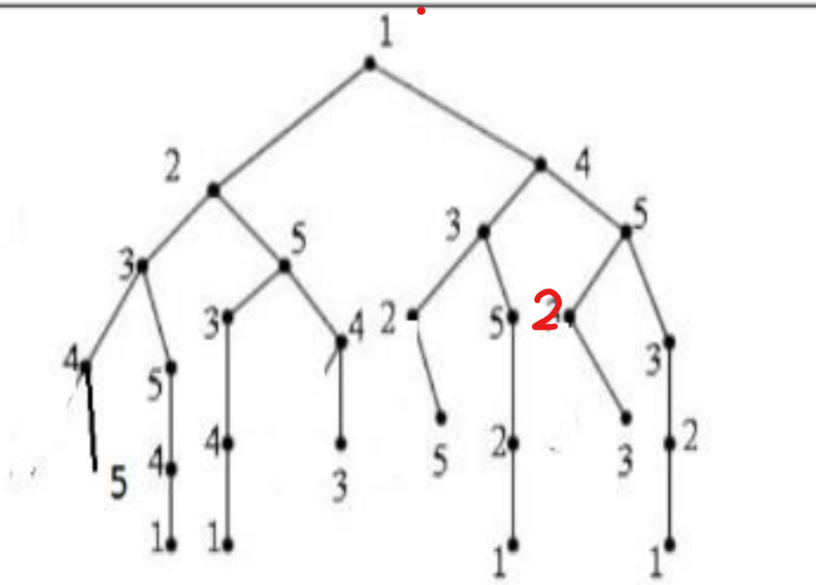
```
LietKeHamilton(u){
    for (v ∈ G) vs[v]= 0;
        x[1]= u; vs[u]= 1;
    Hamilton(2);
}
```

u = 1

Ví dụ 13: Liệt kê chu trình Hamilton

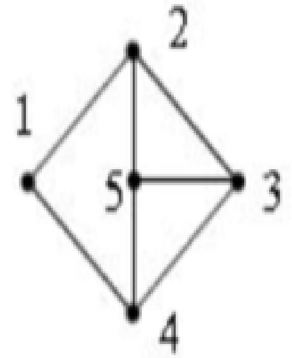


Đồ thị vô hướng G gồm 5 đỉnh



Cây tìm kiếm các chu trình Hamilton của G từ u= 1

Trình bày lời giải dưới dạng bảng:



Đồ thị vô hướng G gồm 5 đỉnh

	1	2	3	4	5
1	0	1	0	1	0
2	1	0	1	0	1
3	0	1	0	1	1
4	1	0	1	0	1
5	0	1	1	1	0

ñ đỉnh nào
có thể quay
lại đỉnh
1

Lập bảng:

Bước	x[1]	x[2]	x[3]	x[4]	x[5]	(x[5],x[1]) ∈ E?
1	1	2	3	4	5	No
2	1	2	3	5	4	Yes
3	1	2	5	3	4	Yes
4	1	2	5	4	3	No
5	1	4	3	2	5	No
6	1	4	3	5	2	Yes
7	1	4	5	2	3	No
8	1	4	5	3	2	Yes

Các chu trình Hamilton của G

- Có tổng cộng 4 chu trình Hamilton tìm được:

(1, 2, 3, 5, 4, 1); (1, 2, 5, 3, 4, 1);

(1, 4, 3, 5, 2, 1); (1, 4, 5, 3, 2, 1);

Ghi chú:

- Một dạng đặc biệt của Bài toán liệt kê chu trình Hamilton là bài toán Người du lịch.
- Cho đến nay, Bài toán liệt kê chu trình Hamilton được coi là bài toán khó và vẫn được tiếp tục nghiên cứu.
- Các thuật toán mã hóa có độ phức tạp tương đương thuật toán liệt kê chu trình Hamilton được xem là chống được tấn công của Hacker và có thể sử dụng an toàn.

Tổng kết chương 3

- **Về lý thuyết**:

- Khái niệm và điều kiện đồ thị Euler, nửa Euler

- Thuật toán tìm chu trình/đường đi Euler

- Khái niệm đồ thị Hamilton, nửa Hamilton

- Thuật toán tìm chu trình/đường đi Hamilton

- **Về các dạng bài tập**

- Viết chương trình mô tả thuật toán

- Tìm chu trình/đường đi Euler và chu trình/đường đi Hamilton.

Thảo luận


THANK YOU