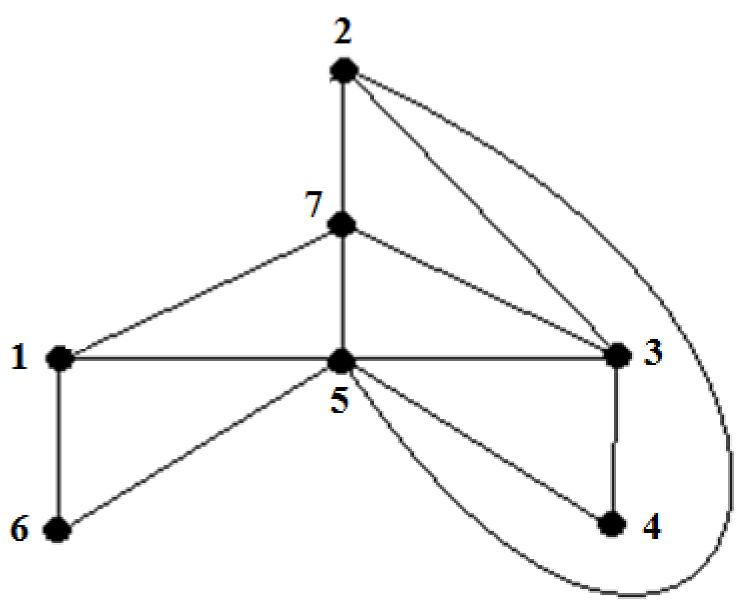
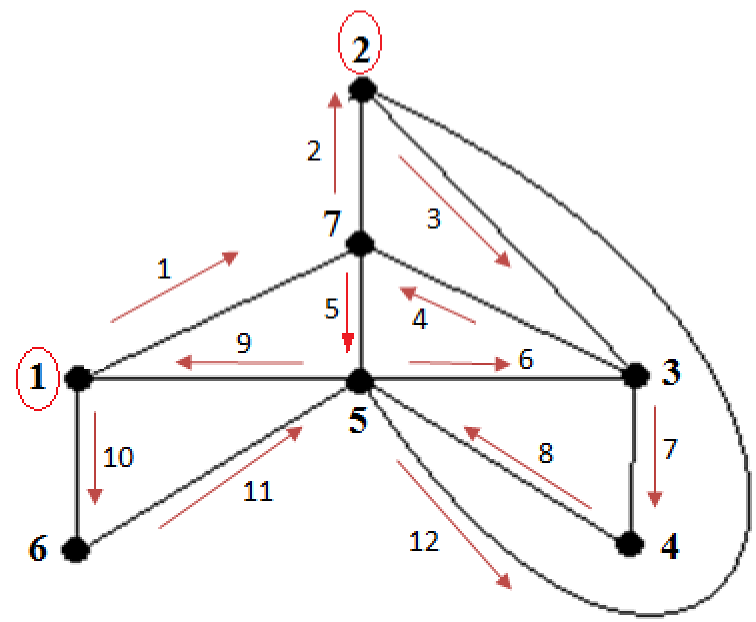
Chương 6.
Đường đi Euler, đường đi hamilton
Chu trình Euler , chu trình Hamilton

6.1 Đường đi Euler, Chu trình Euler :

6.1.1 Định nghĩa : Cho G=(V, E) . Một đường đi Euler trong G là đường đi chứa toàn bộ cạnh của G và mỗi cạnh xuất hiện 1 lần.

Trong đồ thị có hướng đường đi Euler được định nghĩa tương tự.





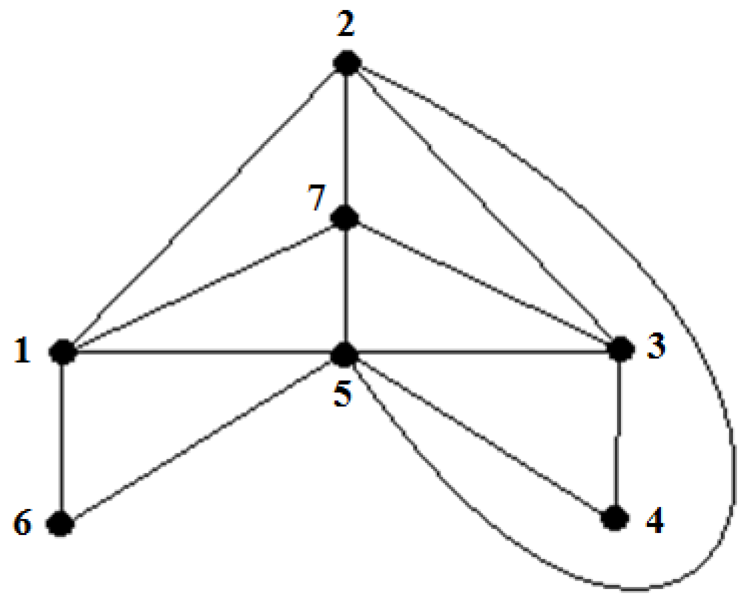
6.1.2 Định nghĩa : Cho G=(V, E) . Một chu trình Euler trong G là chu trình chứa toàn bộ cạnh của G và mỗi cạnh xuất hiện 1 lần.

Trong đồ thị có hướng chu trình Euler được định nghĩa tương tự.

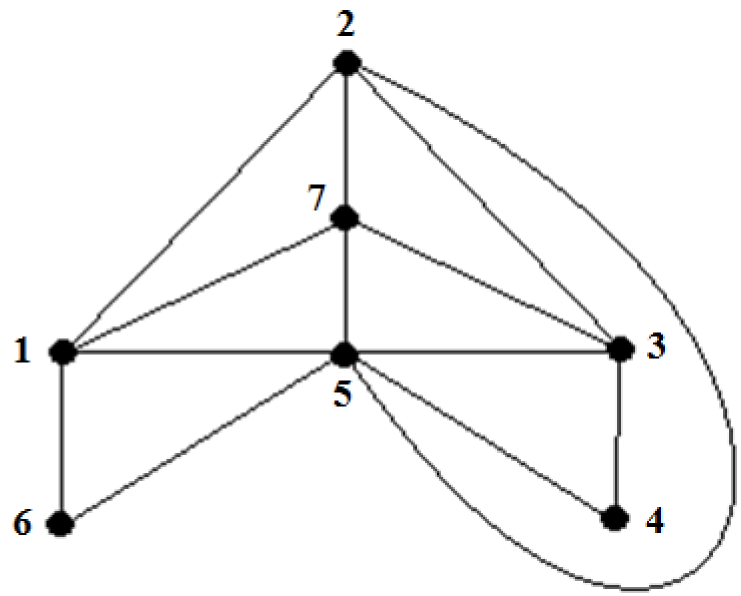
Đồ thị (vô hướng hay có hướng) có chu trình Euler được gọi là đồ thị Euler.

Chú ý : Nếu G chỉ có 1 đỉnh v và không có cạnh ta xem G có một chu trình Euler và ký hiệu (v).

6.1.2 Định lý : Nếu G (vô hướng)có chu trình Euler thì G liên thông và mọi đỉnh đều có bậc chẵn.

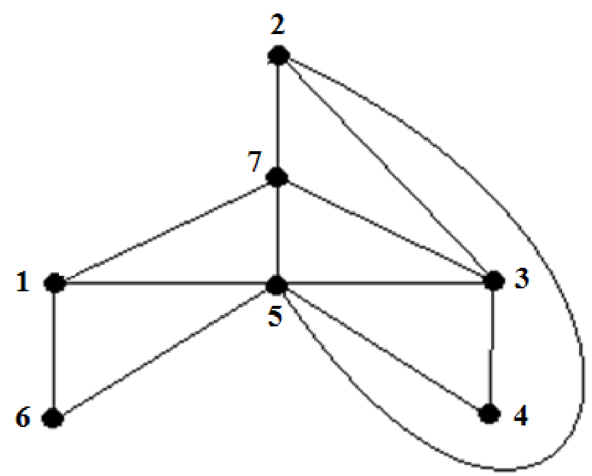


6.1.3 Định lý : Nếu G liên thông và mọi đỉnh của G có bậc chẵn thì G có chu trình E.

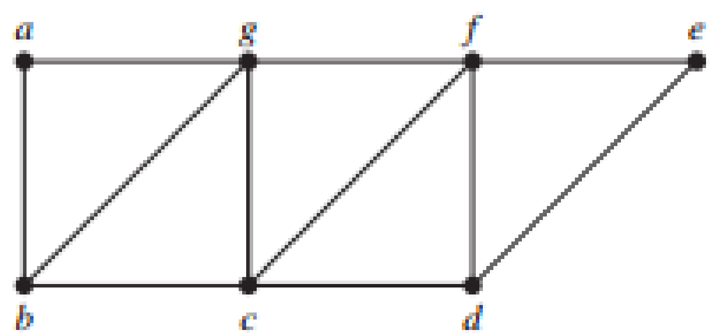


6.1.4 Định lý : Cho G liên thông.

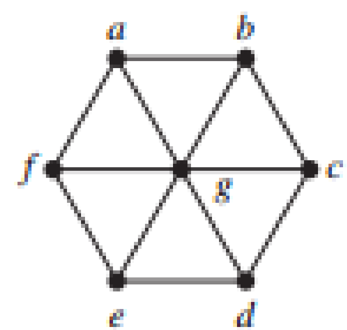
Có đường đi Euler (không phải là chu trình Euler) khi và chỉ khi G có đúng 2 đỉnh có bậc lẻ.



G_1



G_2



G_3

6.1.5 Thuật toán tìm chu trình Euler:

- S : STACK rỗng ban đầu.
- CE : (Stack) tập các cạnh của chu trình Euler, rỗng ban đầu.
- TopStack(S) : cho giá trị phần tử của đỉnh STACK.
- InputStack(S, v) : cho v vào đỉnh STACK.
- DeStack(S) : xóa đỉnh v ở đỉnh STACK.
- Adj(x) : Là *danh sách* các đỉnh kề với đỉnh x.
- Top(Adj(x)) : phần tử đầu của Adj(x).

Bước 1 :

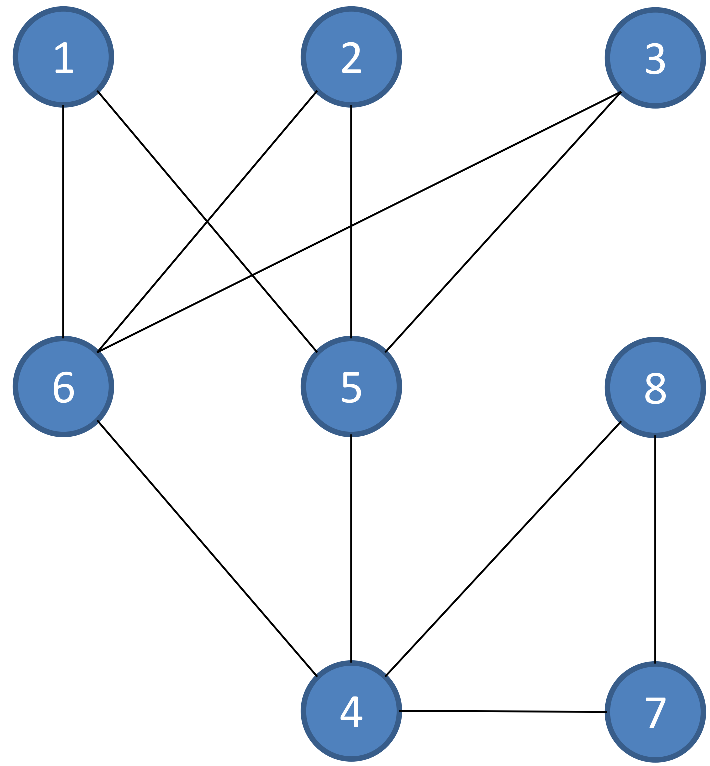
1. **i = 0**

2. u = 3; // u là đỉnh tùy ý.

3. InputStack(S, u);

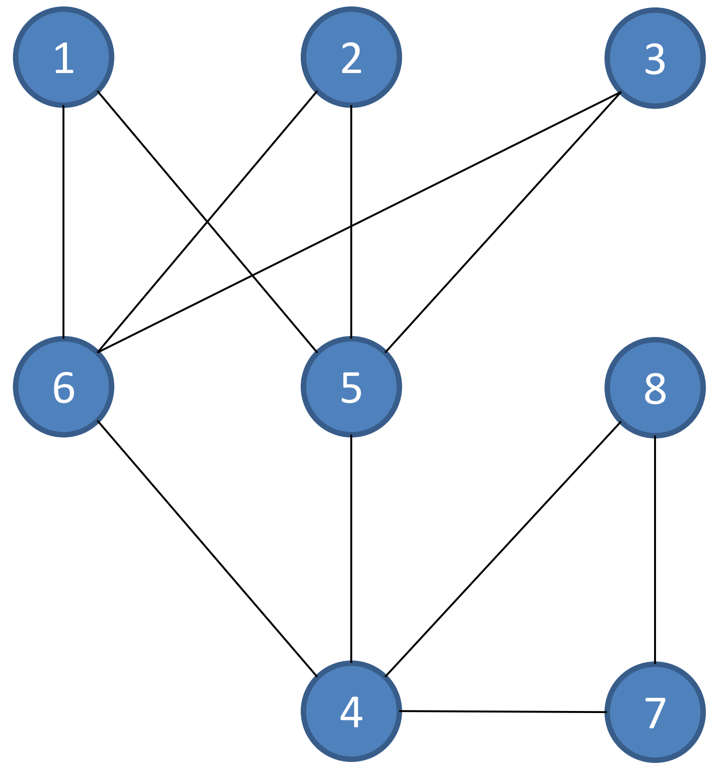
Bước 2 :

```
while (S ≠ ∅)
{    1. x = TopStack(S);
     If (Adj(x) ≠ ∅) {
      2.      y = Top(Adj(x));
      3.      InputStack(S, y);
      4.      Adj(x) = Adj(x) – {y};
      5.      Adj(y) = Adj(y) – {x};
     }
     Else {  6.  InputStack(CE, x);
              7.  DeStack(S);
          }

}
```



u = 3; InputStack(S, u) ; // u là đỉnh tùy ý.

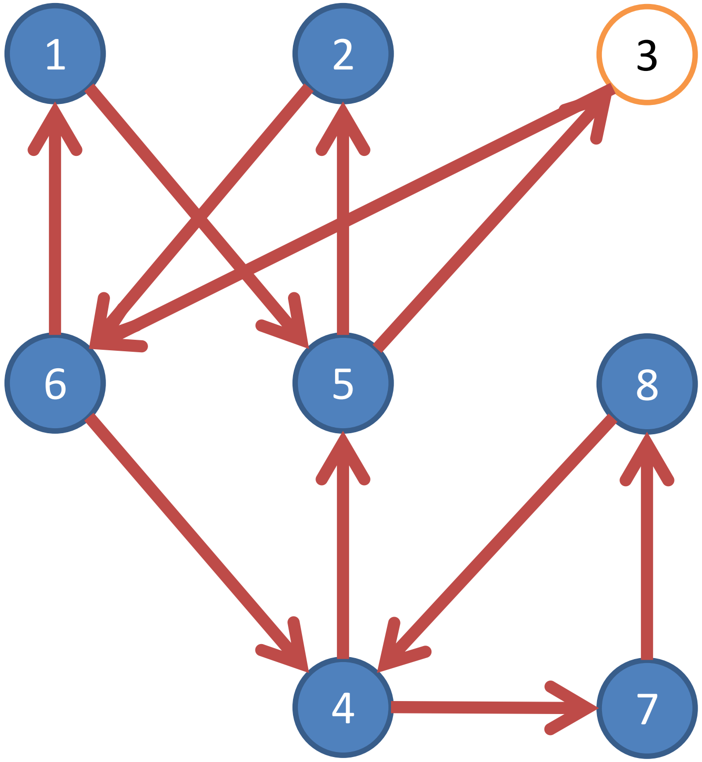
```
while (S ≠ ∅)
{   1. x = TopStack(S);
    If (Adj(x) ≠ ∅) {
     2.     y = Top(Adj(x));
     3.     InputStack(S, y);
     4.     Adj(x) = Adj(x) – {y};
     5.     Adj(y) = Adj(y) – {x};
    }
    Else {  6.  InputStack(CE, x);
            7.  DeStack(S);
         }

}
```



x =

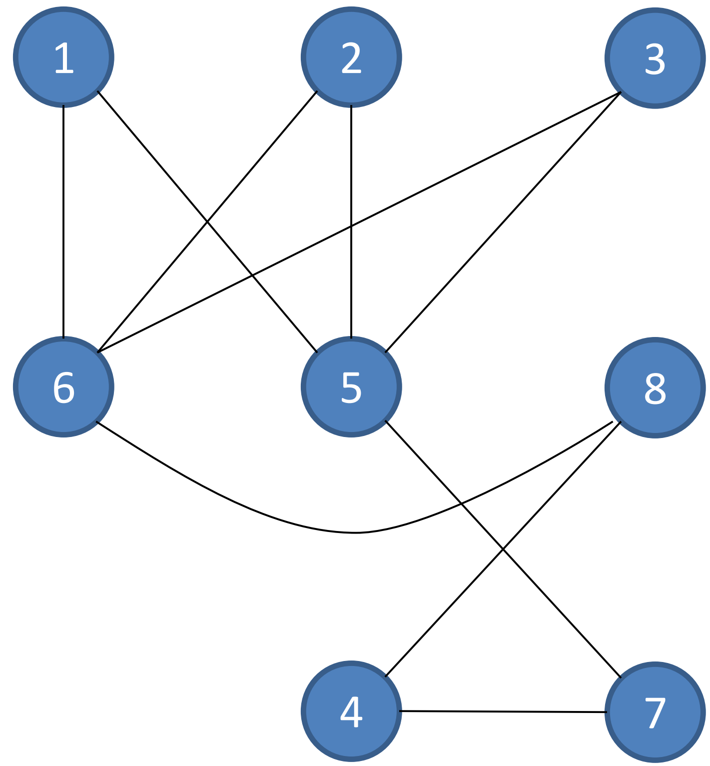
y =

Kết quả :



BÀI TẬP NỘP:

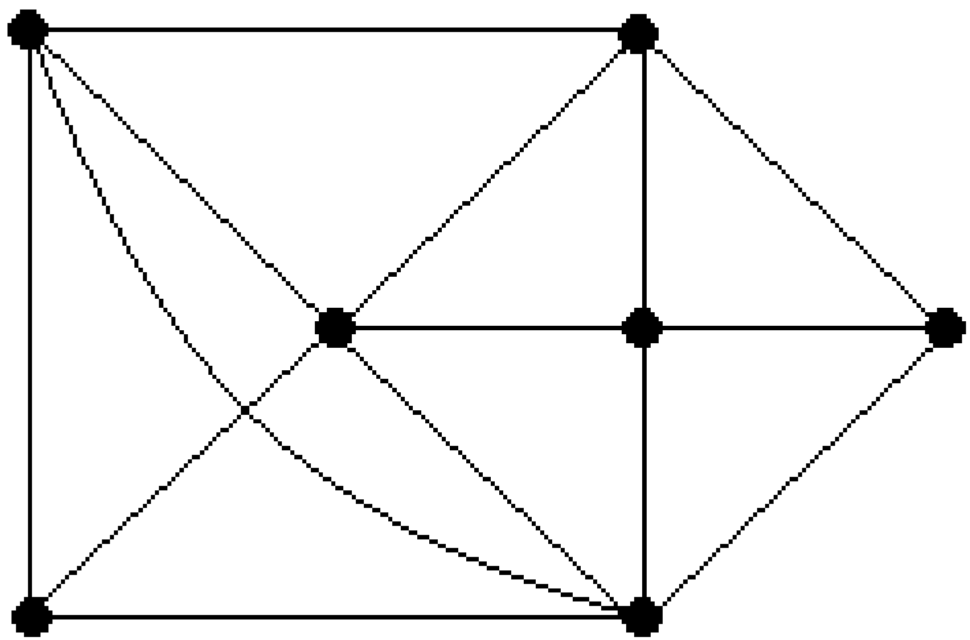
```
u = 3; // u là đỉnh tùy ý.
InputStack(S, u);
while (S ≠ ∅)
{    x = TopStack(S);
     If (Adj(x) ≠ ∅) {
          y = Top(Adj(x));
          InputStack(S, y);
          Adj(x) = Adj(x) – {y};
          Adj(y) = Adj(y) – {x};
     }
     Else {  InputStack(CE, x);
             DeStack(S);
          }
}
```



6.2 Chu trình Hamilton:

6.2.1 Định nghĩa : Cho G=(V, E) là đồ thị vô (có) hướng . Một chu trình sơ cấp trong G chứa tất cả các đỉnh, được gọi là chu trình Hamilton.

Ví dụ :



6.2.2 Định lý Dirac : Cho G=(V, E) là đồ thị vô hướng có n đỉnh ($n \geq 3$). Nếu mọi $v \in V$, $d(v) \geq n / 2$ thì G có chu trình Hamilton.

6.2.3 Định lý Dirac : Cho G=(V, E) là đồ thị có hướng có n đỉnh . Nếu mọi $v \in V$, $d^+(v) \geq n / 2$ và $d^-(v) \geq n / 2$ thì G có chu trình Hamilton.

6.2.3 Thuật toán tìm chu trình Hamilton:

Thủ tục Hamilton(k)

```
{
  For each y ∈ Adj(x_{k-1})
  {   If ((k=n+1) & (y = v_0)) Print(x_1, x_2, . ., x_n, v_0);
      Else If (Color[y]=White)  {
                                 x_k = y;
                                 Color[y]= Gray;
                                 Hamilton(k+1);
                                 Color[y]= White;
                                     }
   }// For
}
```

Main()

{

For each $v \in V$: Color[v]=White;

$x_1 = v_0$;

Color[v_0]=Gray;

Hamilton(2);

}

Ví dụ :