



Cấu Trúc Dữ Liệu & Giải Thuật

Tổng quan Đồ thị



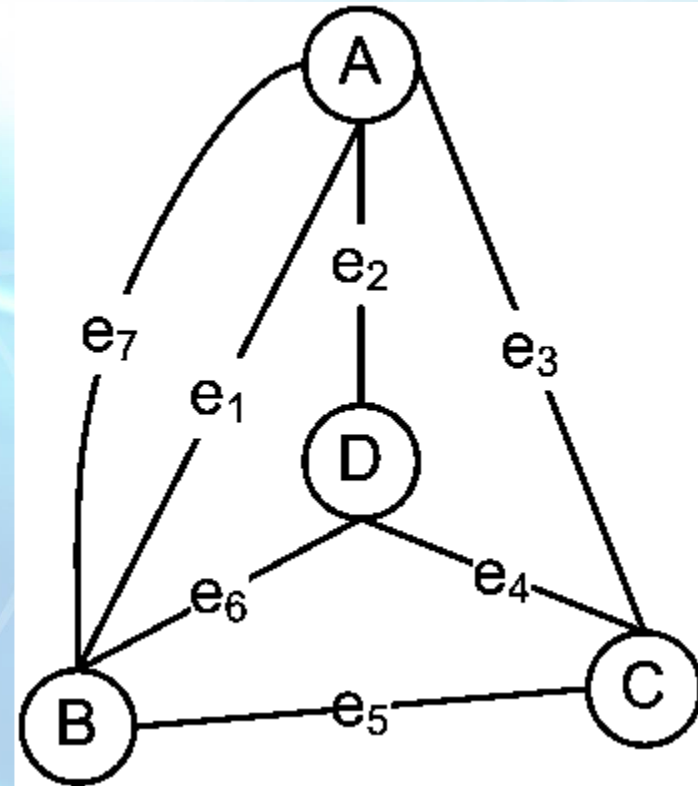
NỘI DUNG

- Đồ thị và các khái niệm trên đồ thị
- Biểu diễn đồ thị trên máy tính
- Duyệt đồ thị theo chiều sâu và chiều rộng
- Một số ứng dụng



ĐỊNH NGHĨA – ĐỒ THỊ CÓ HƯỚNG

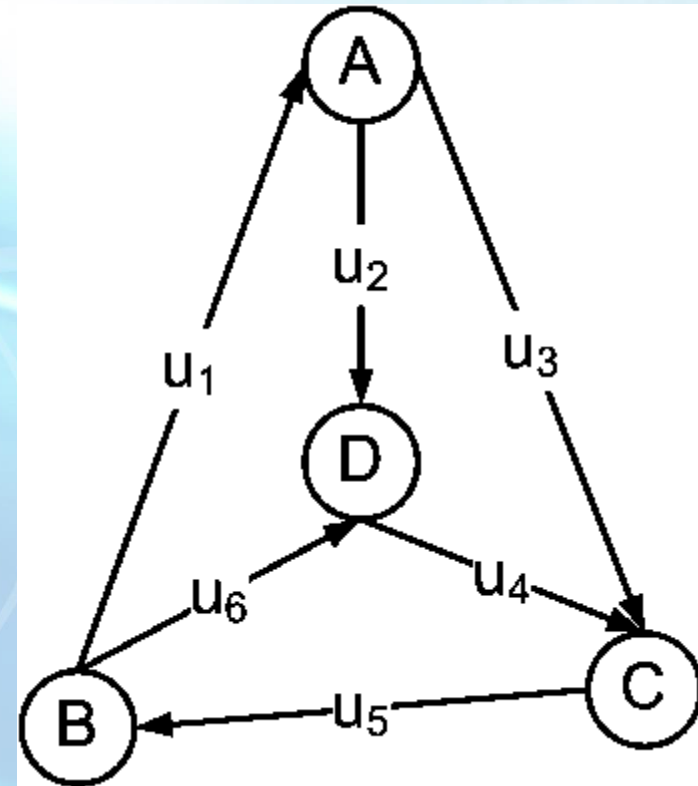
- Một đồ thị vô hướng $G = (V, E)$ được định nghĩa bởi:
 - Tập hợp V được gọi là tập các đỉnh của đồ thị;
 - Tập hợp E là tập các cạnh của đồ thị;
 - Mỗi cạnh $e \in E$ được liên kết với một cặp đỉnh $\{i, j\} \in V^2$,
không phân biệt thứ tự





ĐỊNH NGHĨA – ĐỒ THỊ CÓ HƯỚNG

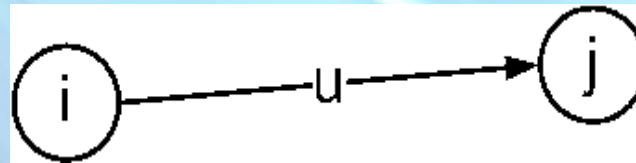
- Một đồ thị có hướng $G = (V, U)$ được định nghĩa bởi:
 - Tập hợp V được gọi là tập các đỉnh của đồ thị;
 - Tập hợp U là tập các cạnh của đồ thị;
 - Mỗi cạnh $u \in U$ được liên kết với một cặp đỉnh $(i, j) \in V^2$.





ĐỈNH KỀ

- Trên đồ thị có hướng, xét cạnh u được liên kết với cặp đỉnh (i, j) :

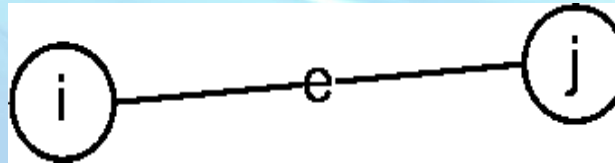


- Cạnh u **kề** với đỉnh i và đỉnh j (hay đỉnh i và đỉnh j **kề** với cạnh u); có thể viết tắt $u = (i, j)$. Cạnh u đi ra khỏi đỉnh i và đi vào đỉnh j
- Đỉnh j được gọi là đỉnh kề của đỉnh i



ĐỈNH KỀ

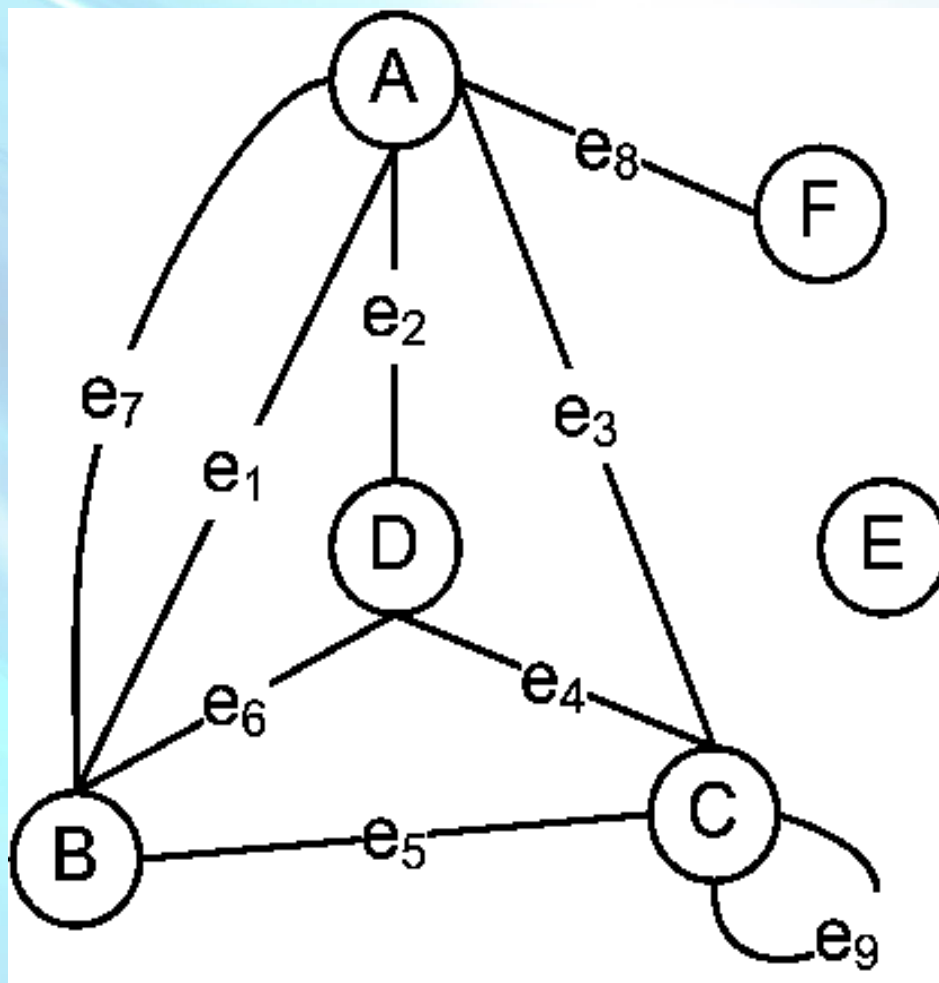
- Trên đồ thị vô hướng, xét cạnh e được liên kết với cặp đỉnh (i, j) :



- Cạnh e **kề** với đỉnh i và đỉnh j (hay đỉnh i và đỉnh j **kề** với cạnh e); có thể viết tắt $e = (i, j)$.
- Đỉnh i và đỉnh j được gọi là 2 đỉnh kề nhau (hay đỉnh i kề với đỉnh j và ngược lại, đỉnh j kề với đỉnh i)



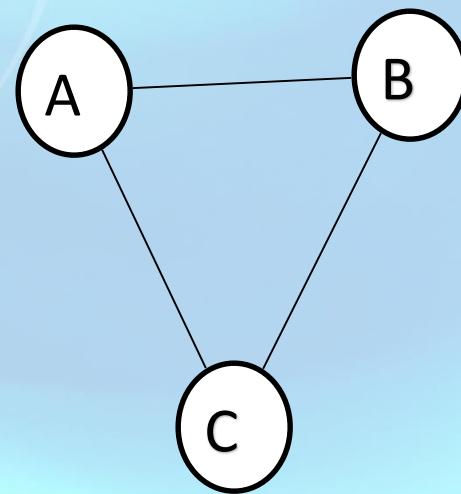
KHUYÊN, ĐỈNH TREO, ĐỈNH CÔ LẬP





CÁC DẠNG ĐỒ THỊ

- Đồ thị RỖNG: tập cạnh là tập rỗng
- Đồ thị ĐƠN: không có khuyên và cạnh song song
- Đồ thị ĐỦ: đồ thị vô hướng, đơn, giữa hai đỉnh bất kỳ đều có đúng một cạnh.
 - Đồ thị đủ N đỉnh ký hiệu là K_N .
 - K_N có $N(N - 1)/2$ cạnh.



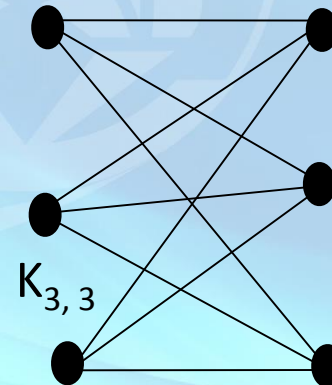
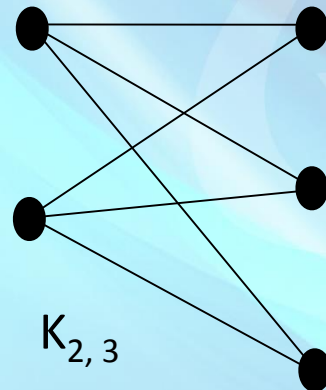
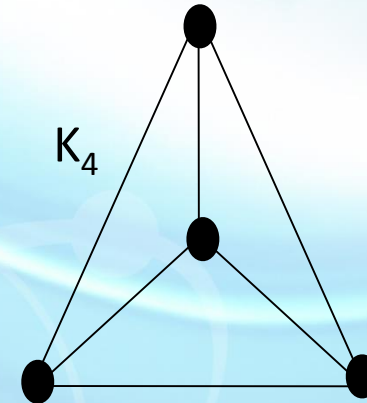
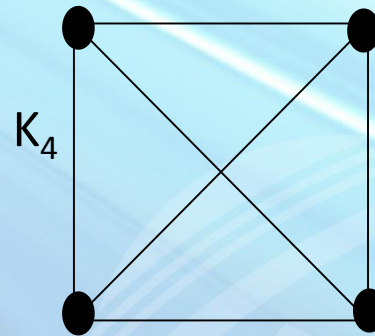
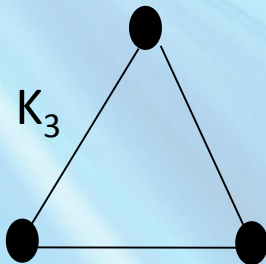


CÁC DẠNG ĐỒ THỊ

- Đồ thị LƯỠNG PHÂN: đồ thị $G = (X, E)$ được gọi là đồ thị lưỡng phân nếu tập X được chia thành hai tập X_1 và X_2 thỏa:
 - X_1 và X_2 phân hoạch;
 - Cạnh chỉ nối giữa X_1 và X_2 .
- Đồ thị LƯỠNG PHÂN ĐỦ: là đồ thị lưỡng phân đơn, vô hướng thỏa với $\forall(i, j): i \in X_1$ và $j \in X_2$ có đúng một cạnh i và j .



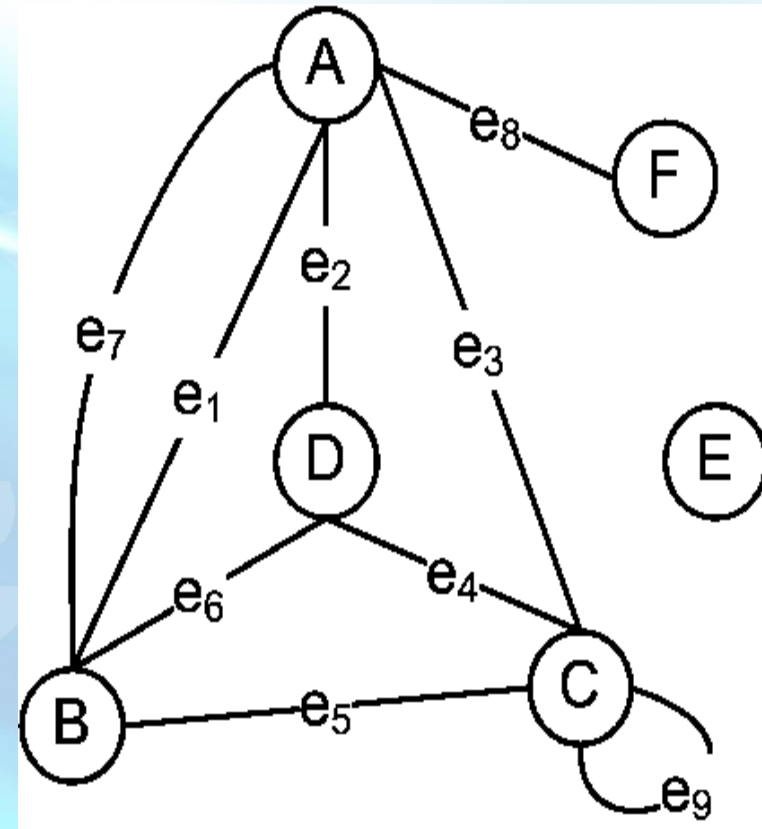
VÍ DỤ ĐỒ THỊ ĐẦY ĐỦ





BẬC CỦA ĐỈNH

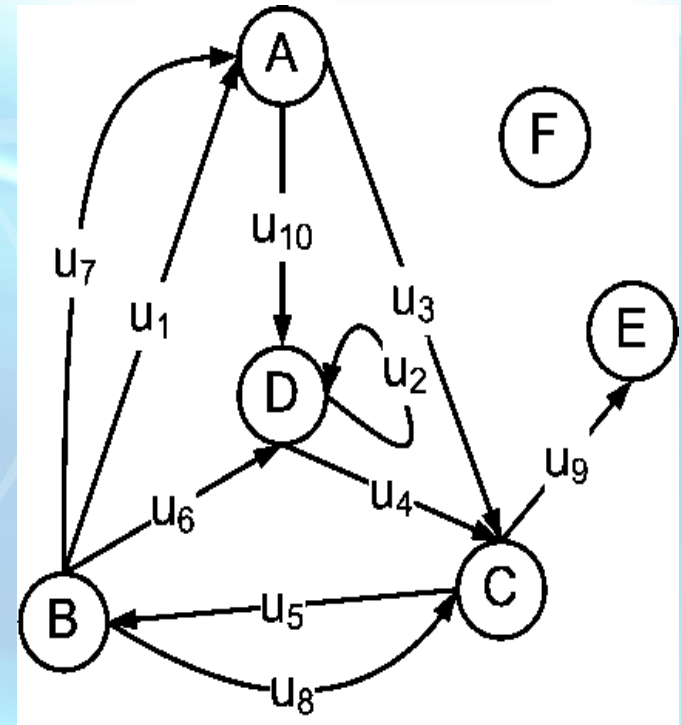
- Xét đồ thị vô hướng G
 - Bậc của đỉnh x trong đồ thị G là số các cạnh kề với đỉnh x , mỗi khuyên được tính hai lần
 - Ký hiệu: $dG(x)$ (hay $d(x)$) nếu đang xét một đồ thị nào đó.





BẬC CỦA ĐỒ THỊ

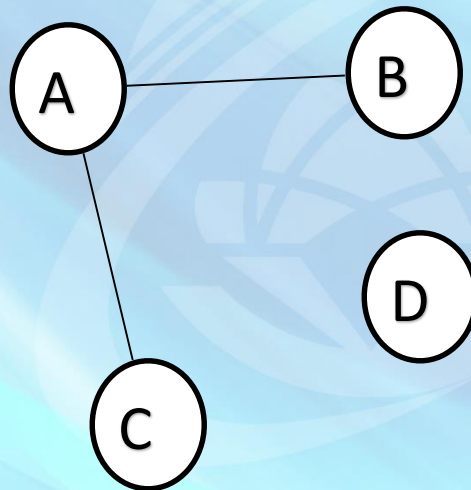
- Xét đồ thị có hướng G
 - **Nửa bậc ngoài** của đỉnh x là số các cạnh đi ra khỏi đỉnh x , ký hiệu $d^+(x)$.
 - **Nửa bậc trong** của đỉnh x là số các cạnh đi vào đỉnh x , ký hiệu $d^-(x)$.
 - **Bậc** của đỉnh x : $d(x) = d^+(x) + d^-(x)$





BẬC CỦA ĐỈNH

- Đỉnh TREO là đỉnh có bậc bằng 1.
- Đỉnh CÔ LẬP là đỉnh có bậc bằng 0.





MỐI LIÊN HỆ BẬC – SỐ CẠNH

- Định lý:

- Xét đồ thị có hướng $G = (X, U)$. Ta có:

$$\sum_{x \in X} d^+(x) = \sum_{x \in X} d^-(x) \text{ và } \sum_{x \in X} d(x) = 2|U|$$

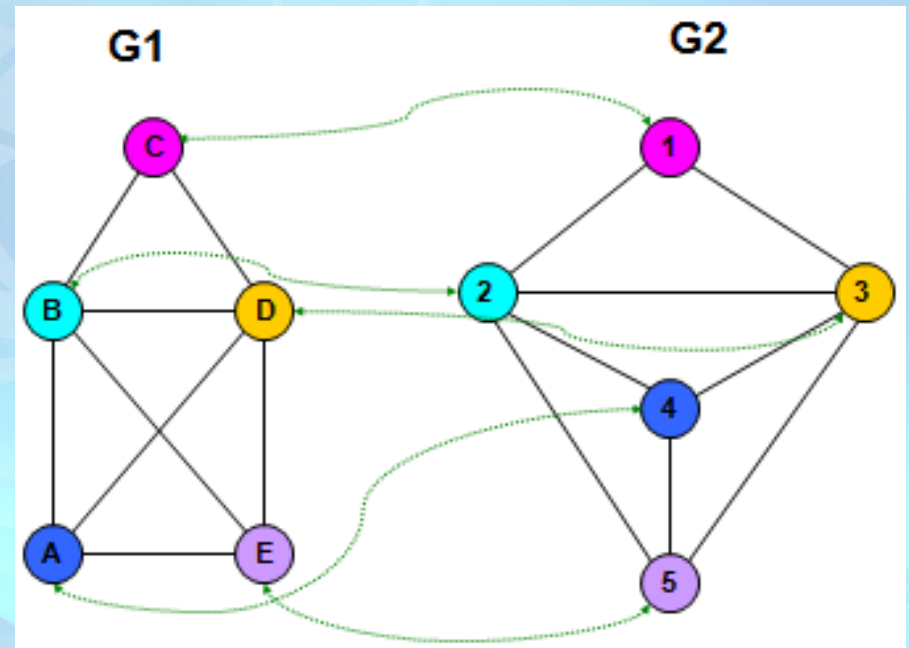
- Xét đồ thị vô hướng $G = (X, E)$. Ta có:

$$\sum_{x \in X} d(x) = 2|E|$$



ĐẲNG CẤU ĐỒ THỊ

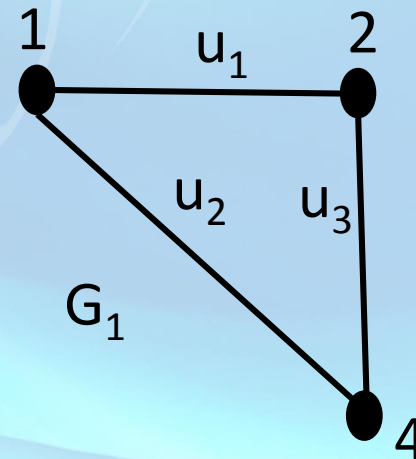
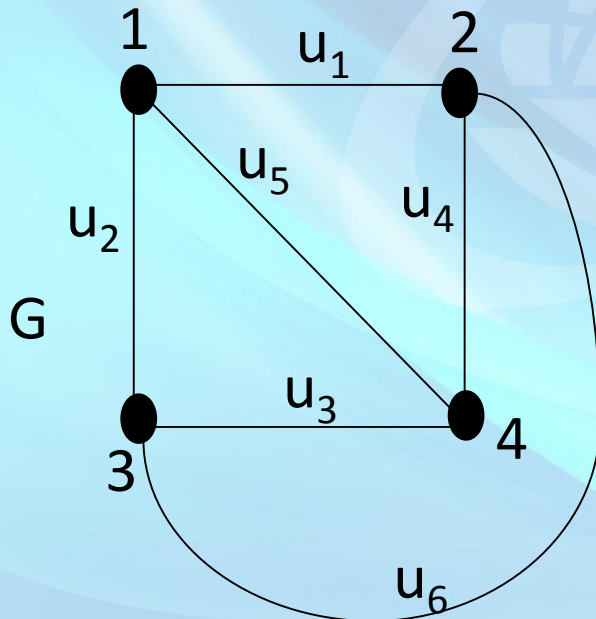
- Hai đồ thị vô hướng $G_1 = (X_1, U_1)$ và $G_2 = (X_2, U_2)$ được gọi là đẳng cấu với nhau nếu :
 - Có cùng số đỉnh.
 - Có cùng số đỉnh bậc k , mọi k nguyên dương ≥ 0 .
 - Cùng số cạnh.
 - Cùng số thành phần.





ĐỒ THỊ CON

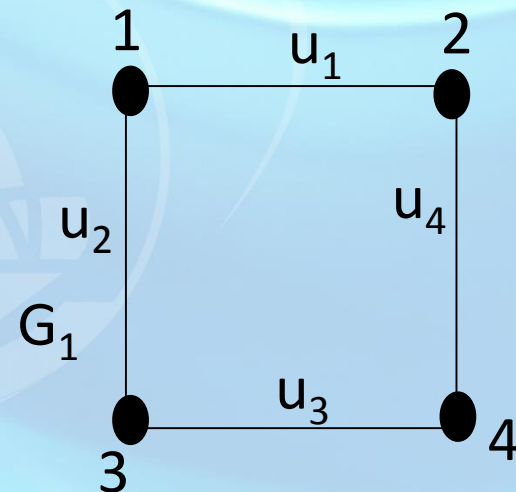
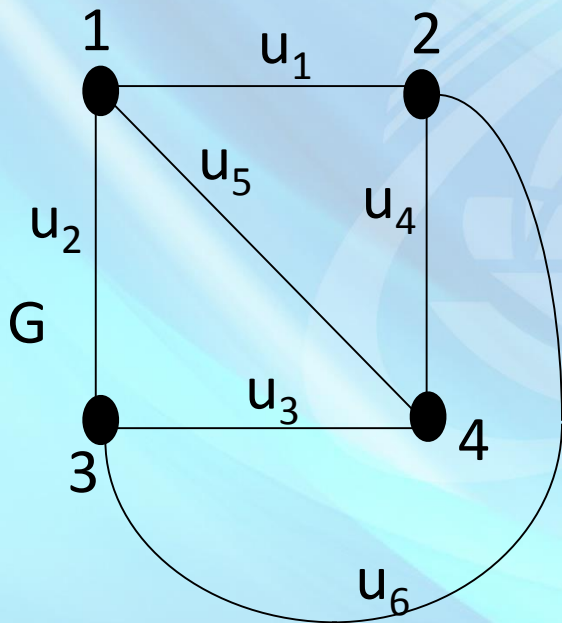
- Xét hai đồ thị $G = (X, U)$ và $G_1 = (X_1, U_1)$. G_1 được gọi là đồ thị con của G và ký hiệu $G_1 \in G$ nếu:
 - $X_1 \in X; U_1 \in U$
 - $u = (i, j) \in U$ của G , nếu $u \in U_1$ thì $i, j \in X_1$





ĐỒ THỊ BỘ PHẬN

- Đồ thị con $G_1 = (X_1, U_1)$ của đồ thị $G = (X, U)$ được gọi là đồ thị bộ phận của G nếu $X = X_1$.





DÂY CHUYỀN, CHU TRÌNH

- Một dây chuyền trong $G = (X, U)$ là một đồ thị con $C = (V, E)$ của G với:
 - $V = \{x_1, x_2, \dots, x_M\}$
 - $E = \{u_1, u_2, \dots, u_{M-1}\}$ với $u_1 = x_1x_2, u_2 = x_2x_3, \dots, u_{M-1} = x_{M-1}x_M$; liên kết $x_i x_{i+1}$ không phân biệt thứ tự
- Khi đó, x_1 và x_M được nối với nhau bằng dây chuyền C . x_1 là đỉnh đầu và x_M là đỉnh cuối của C .
- Số cạnh của C được gọi là độ dài của C .
- Khi các cạnh hoàn toàn xác định bởi cặp đỉnh kề, dây chuyền có thể viết gọn (x_1, x_2, \dots, x_M)



DÂY CHUYỀN, CHU TRÌNH

- Dây chuyền SƠ CẤP: dây chuyền không có đỉnh lặp lại.
- CHU TRÌNH: là một dây chuyền có đỉnh đầu và đỉnh cuối trùng nhau.



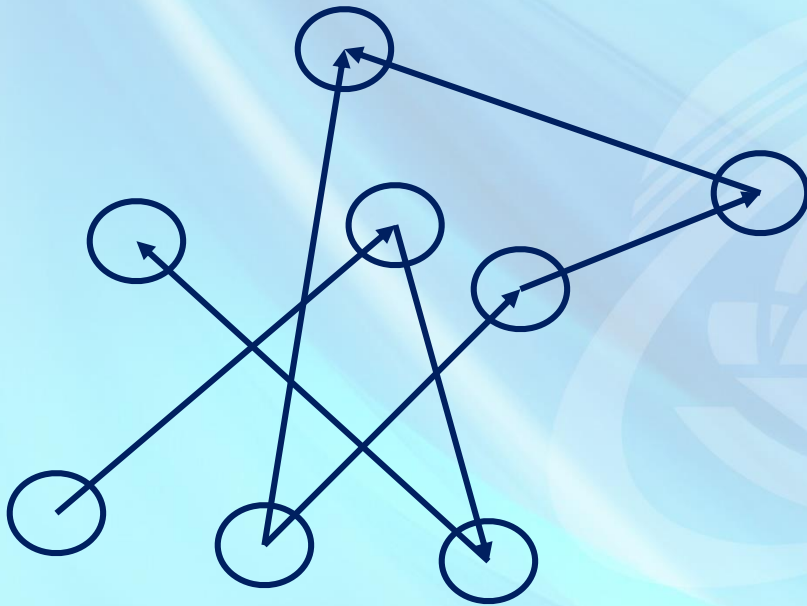
THÀNH PHẦN LIÊN THÔNG

- Một **thành phần liên thông** của một đồ thị vô hướng là một đồ thị con trong đó:
 - Giữa bất kì hai đỉnh nào đều có đường đi đến nhau, và
 - Không thể nhận thêm bất kì một đỉnh nào mà vẫn duy trì tính chất trên.

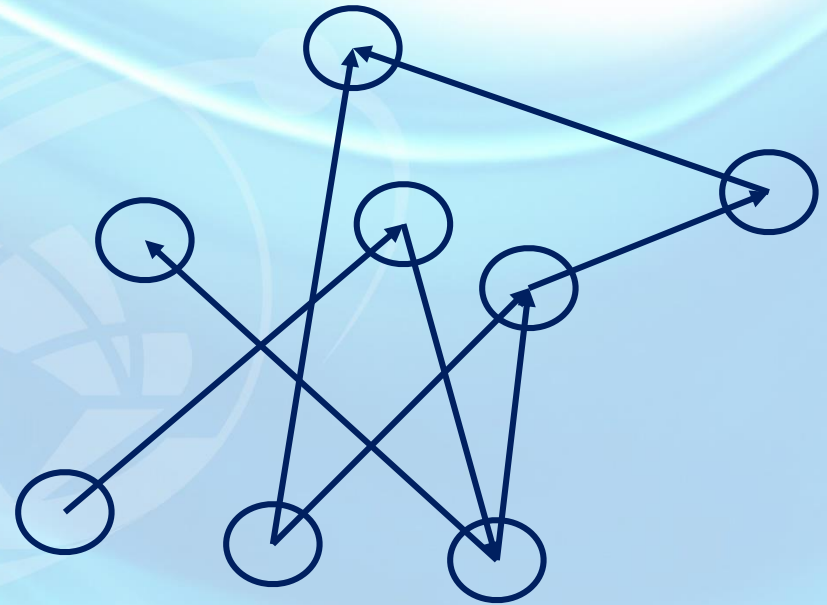


THÀNH PHẦN LIÊN THÔNG

- G gồm 2 thành phần liên thông, H là đồ thị liên thông



G



H



THÀNH PHẦN LIÊN THÔNG

Thuật toán xác định các thành phần liên thông

Input: đồ thị $G = (X, E)$, tập X gồm N đỉnh $1, 2, \dots, N$

Output: các đỉnh của G được gán nhãn là số hiệu của thành phần liên thông tương ứng

1. Khởi tạo biến $label = 0$ và gán nhãn 0 cho tất cả các đỉnh
2. Duyệt qua tất cả các đỉnh $i \in X$

Nếu nhãn của i là 0

1. $label = label + 1$
2. Gán nhãn cho tất cả các đỉnh cùng thuộc thành phần liên thông với i là $label$



THÀNH PHẦN LIÊN THÔNG

Thuật toán gán nhãn các đỉnh cùng thuộc thành phần liên thông với đỉnh i – $\text{Visit}(i, \text{label})$

Input: đồ thị $G = (X, E)$, đỉnh i , nhãn label

Output: các đỉnh cùng thuộc thành phần liên thông với i được gán nhãn label

1. Gán nhãn label cho đỉnh i
2. Duyệt qua tất cả các đỉnh $j \in X$ và có cạnh nối với i

Nếu nhãn của j là 0

$\text{Visit}(j, \text{label})$

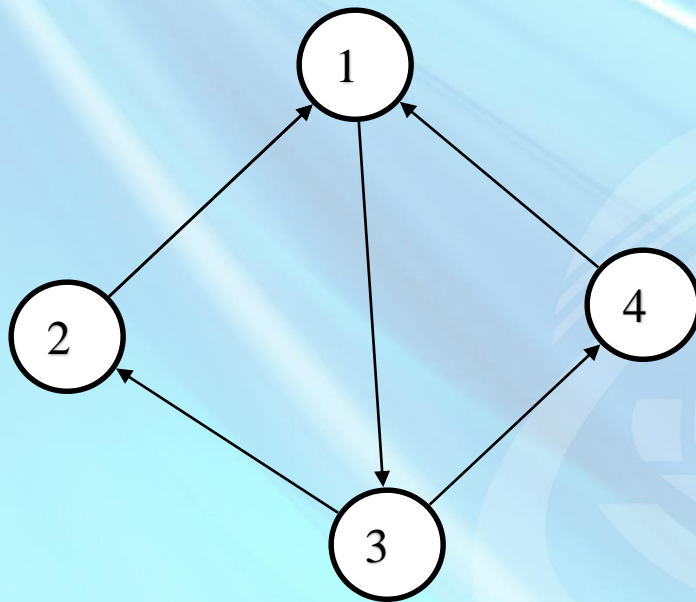


BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN

- Ma trận KỀ:
 - Xét đồ thị $G = (X, U)$, giả sử tập X gồm N đỉnh và được sắp thứ tự $X = \{x_1, x_2, \dots, x_N\}$, tập U gồm M cạnh và được sắp thứ tự $U = \{u_1, u_2, \dots, u_M\}$.
 - Ma trận kề của đồ thị G , ký hiệu $B(G)$, là một ma trận nhị phân cấp $N \times N$: $B = (B_{ij})$ với B_{ij} được định nghĩa:
 - $B_{ij} = 1$ nếu có cạnh nối x_i tới x_j ,
 - $B_{ij} = 0$ trong trường hợp ngược lại.



BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ



G

$$B = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

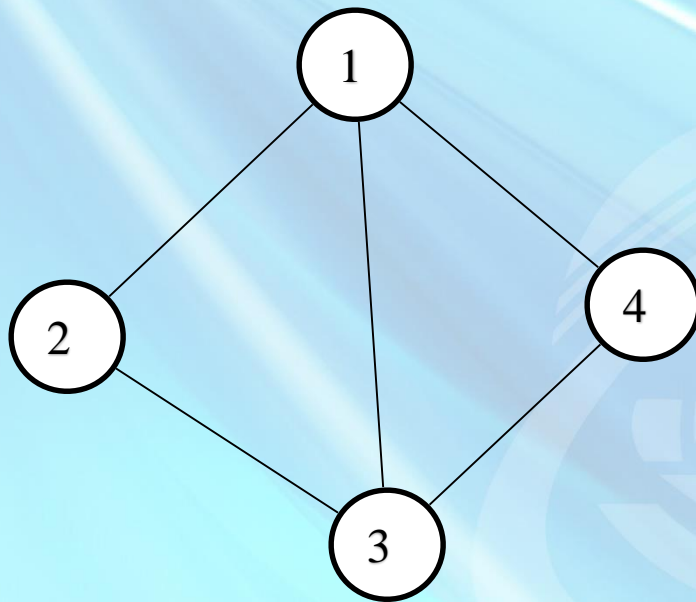


BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN

- Ma trận của đồ thị vô hướng:
 - Xét đồ thị $G = (X, U)$ vô hướng, giả sử tập X gồm N đỉnh và được sắp thứ tự $X = \{x_1, x_2, \dots, x_N\}$, tập U gồm M cạnh và được sắp thứ tự $U = \{u_1, u_2, \dots, u_M\}$.
 - Ma trận của G , ký hiệu $A(G)$, là ma trận nhị phân $N \times M$: $A = (A_{ij})$ với A_{ij} được định nghĩa:
 - $A_{ij} = 1$ nếu đỉnh x_i kề với cạnh u_j ,
 - $A_{ij} = 0$ nếu ngược lại.



BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ

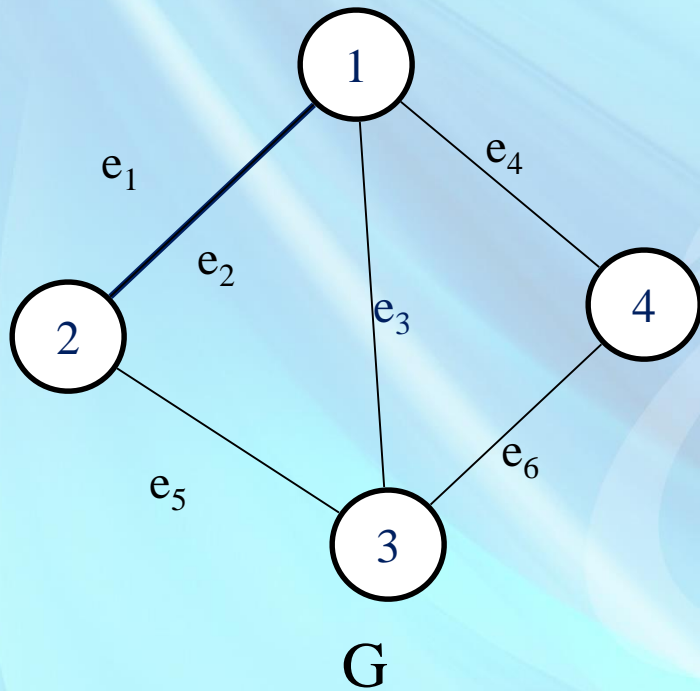


G

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$



BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN TRỌNG SỐ





CÀI ĐẶT BẰNG NGÔN NGỮ C++

```
#define MaxV 20 // số đỉnh cực đại của đồ thị
```

```
int A[MaxV][MaxV]; //Ma trận kề
```

```
int ChuaXet[MaxV]; // sử dụng xét thành phần  
liên thông
```



DUYỆT THEO CHIỀU CHIỀU SÂU - DFS

- Thuật toán duyệt theo chiều sâu (Deep-First Search-DFS)
 - Cho $G = (V, E)$ là đồ thị có tập các đỉnh V và tập các cạnh E . v là một đỉnh trong V và u là đỉnh kề của v , sao cho u cũng thuộc V .
 - Khi đó ta dán nhãn cho tất cả các đỉnh của đồ thị là 0. Chọn một đỉnh v thuộc tập V để bắt đầu duyệt. Gán nhãn đỉnh v này là 1: v đã được duyệt.
 - Chọn đỉnh u trong tập V kề với đỉnh v mà nhãn là 0. Duyệt qua đỉnh u và gán nhãn u là 1. Tiếp tục quá trình duyệt đến khi tất cả các đỉnh đồ thị có nhãn là 1



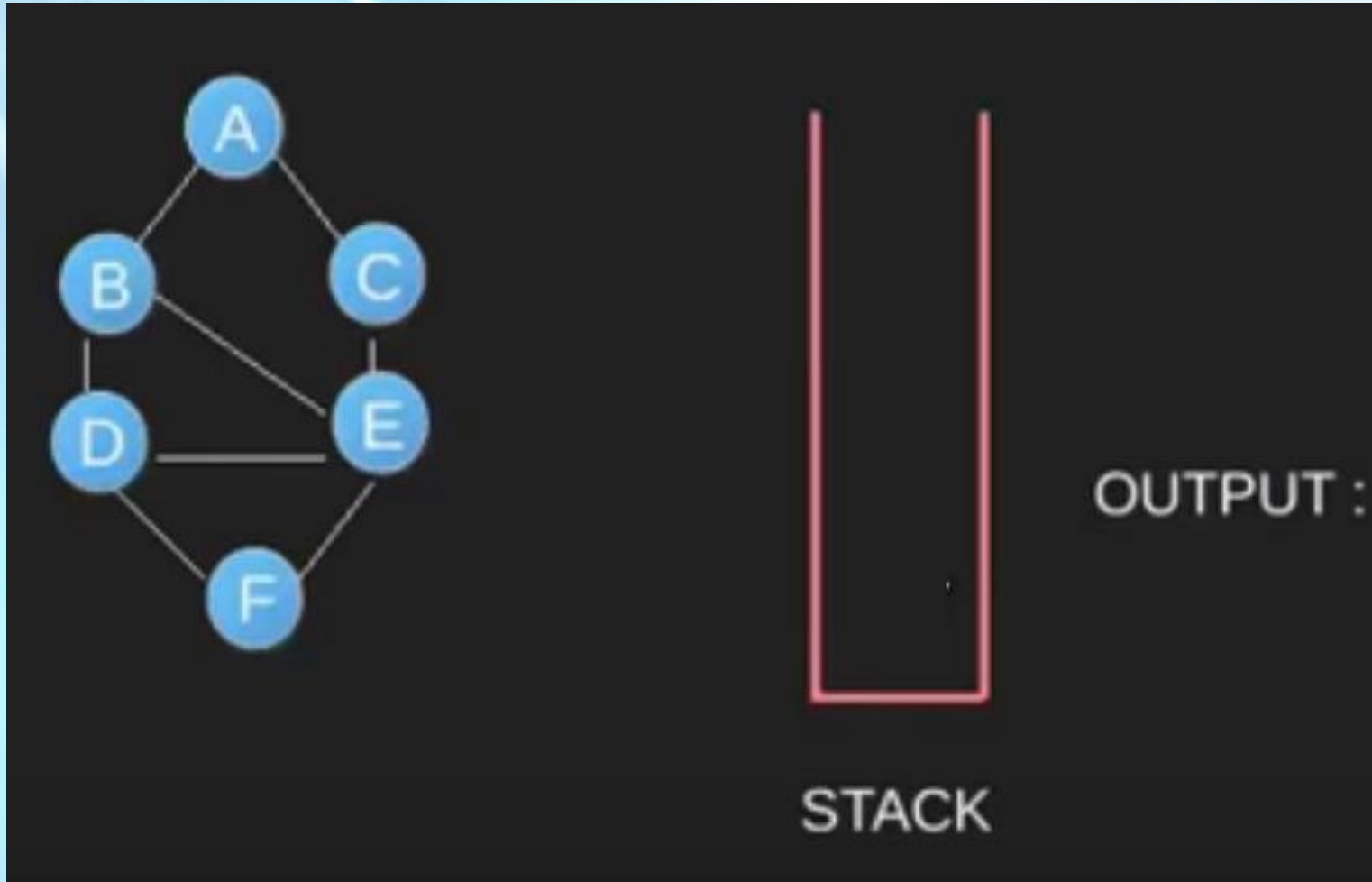
DUYỆT ĐỒ THỊ

Void DFS (int v)

```
{  
    Gắn nhãn v đã duyệt;  
    for (u = 1; u <= n; u++)  
        if (u tồn tại trong danh sách kề V)  
            if(u có nhãn là 0)  
            {  
                Xử lý đỉnh u; //Gắn nhãn 1  
                DFS (u);  
            }  
}
```

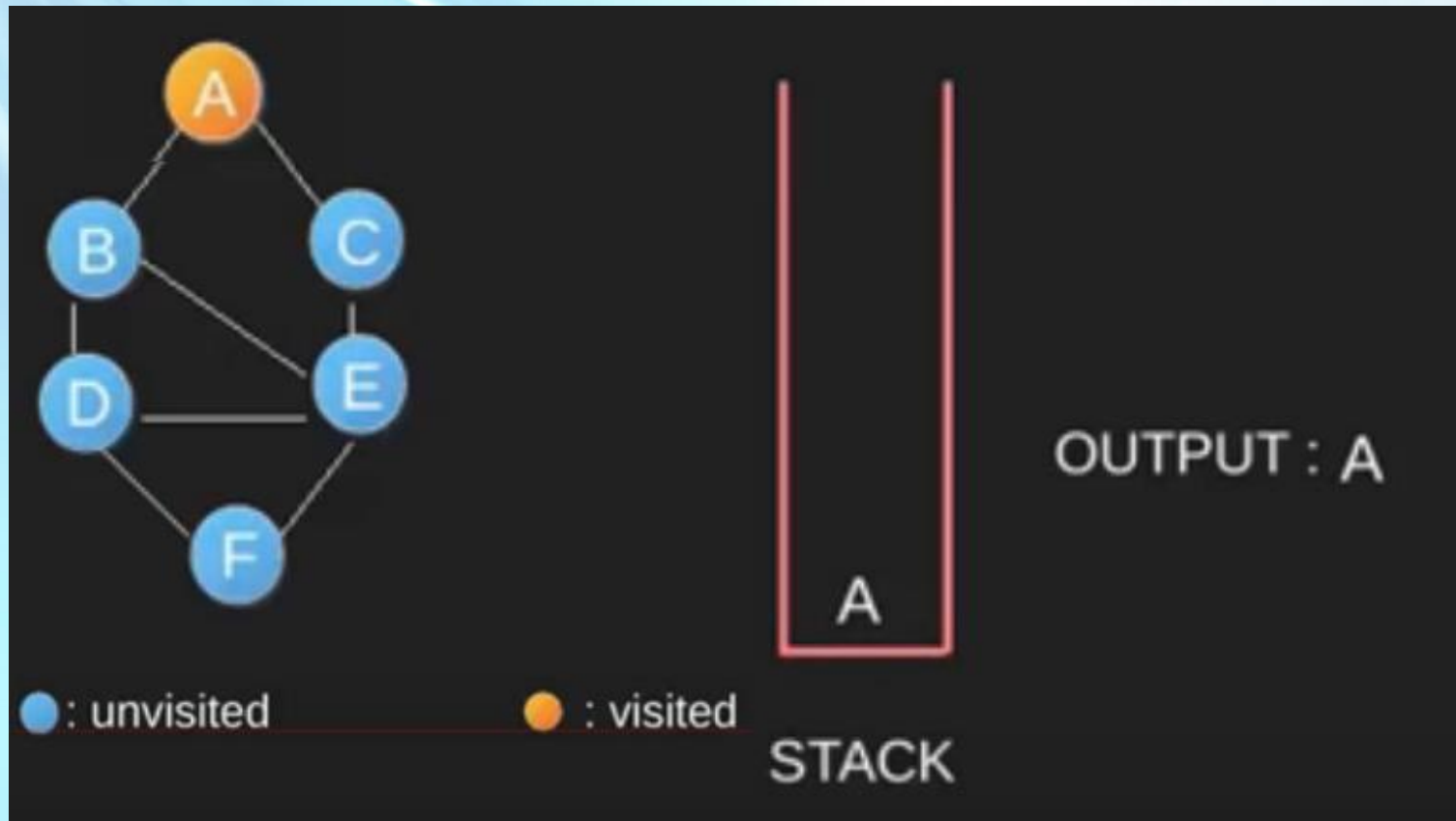


DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



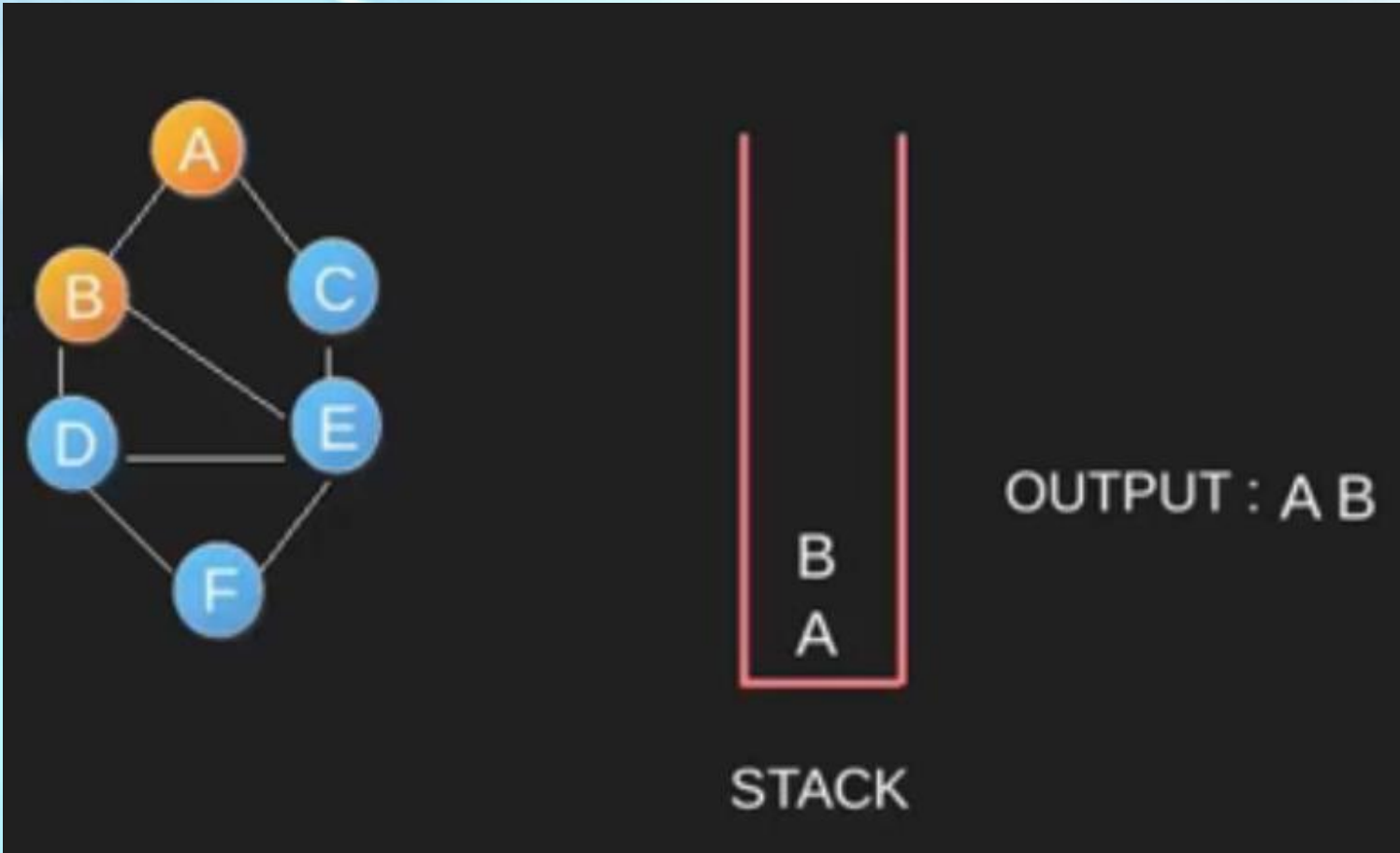


DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



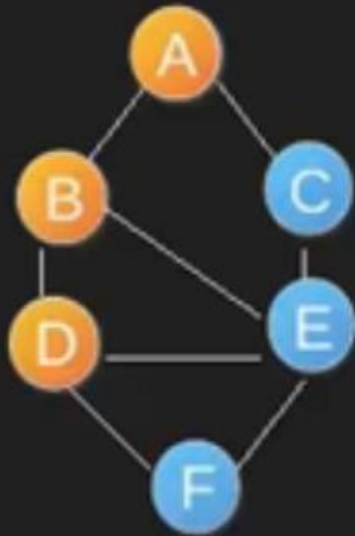


DUYỆT THEO CHIỀU CHIỀU SÂU - DFS





DUYỆT THEO CHIỀU CHIỀU SÂU - DFS

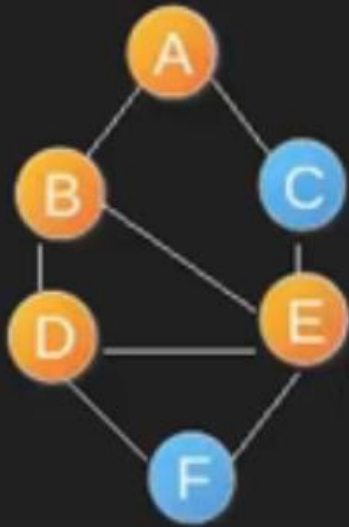


STACK

OUTPUT : A B D



DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



Note : E is a neighbour
of D

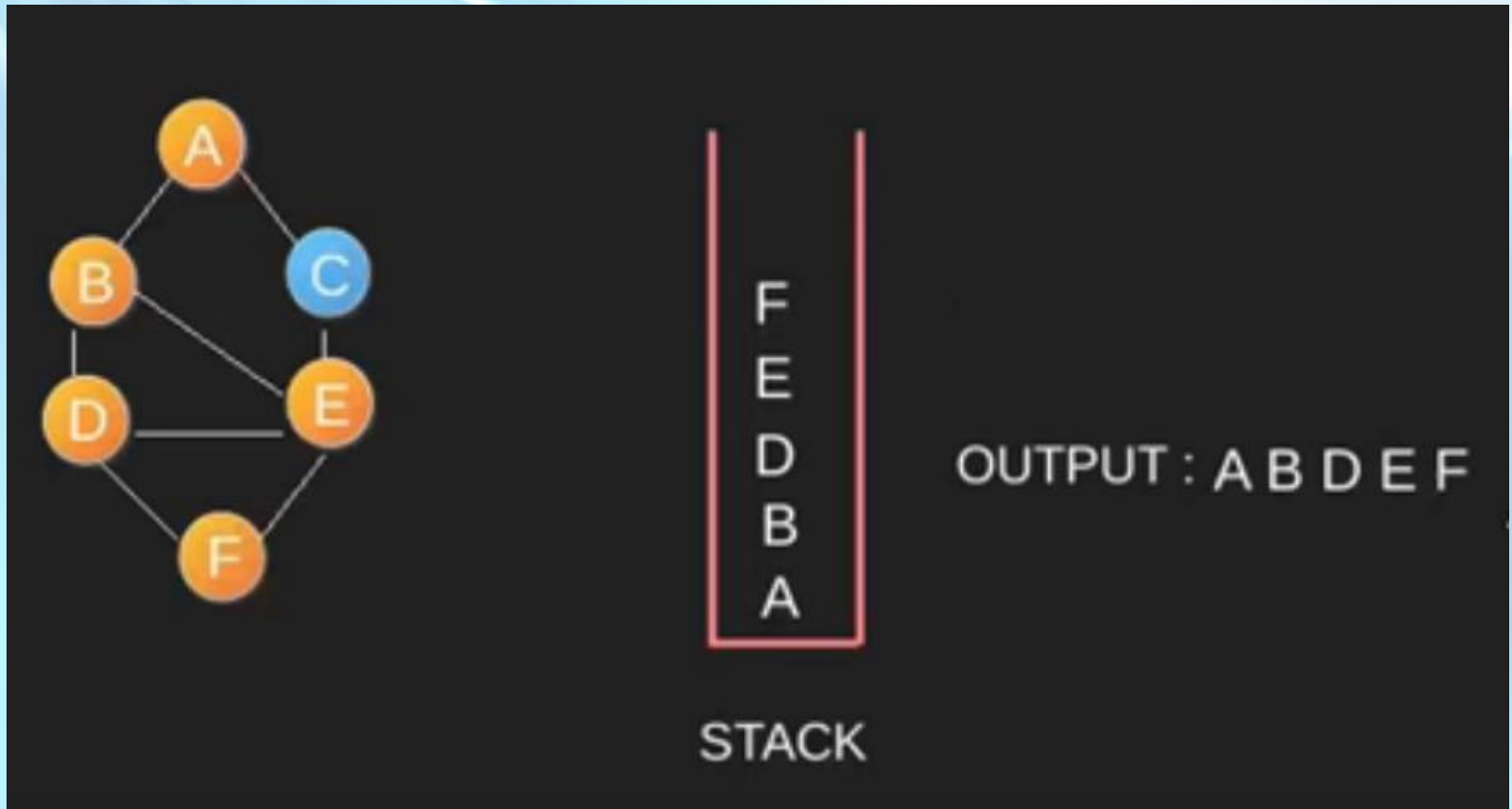
E
D
B
A

STACK

OUTPUT : A B D E

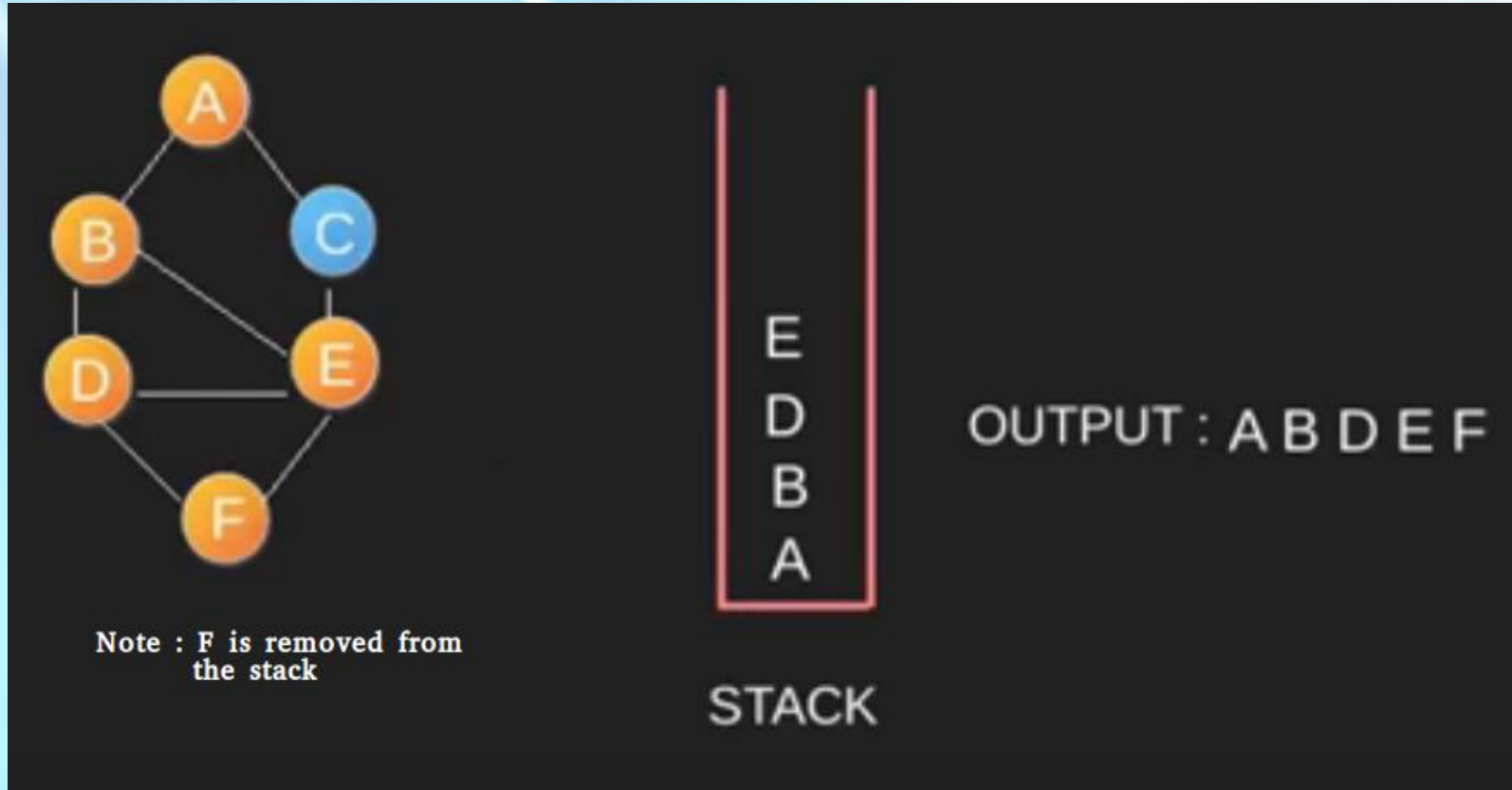


DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



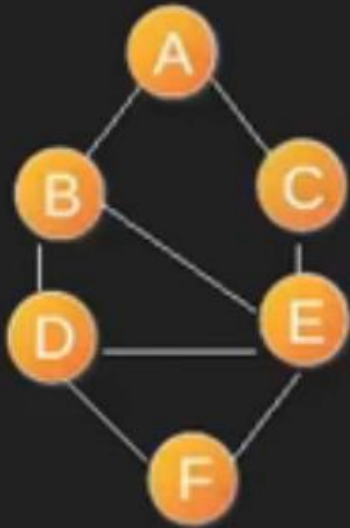


DUYỆT THEO CHIỀU CHIỀU SÂU - DFS





DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



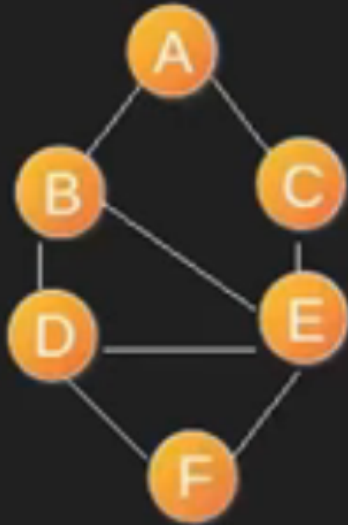
C
E
D
B
A

STACK

OUTPUT : A B D E F C



DUYỆT THEO CHIỀU CHIỀU SÂU - DFS



C, E, D, B and A are one by one removed from stack. Since all nodes are visited, no more nodes are added.



STACK

OUTPUT : A B D E F C



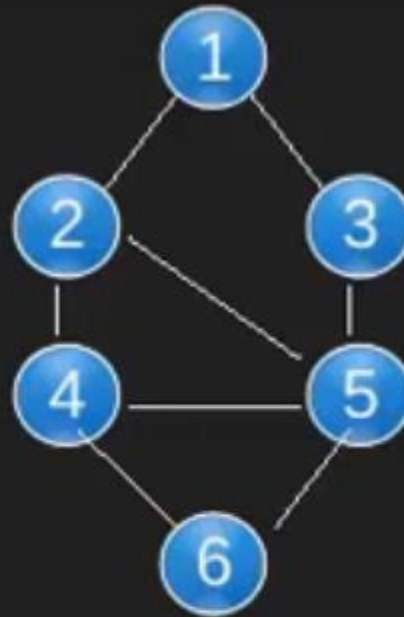
DUYỆT THEO CHIỀU RỘNG - BFS

Thuật toán duyệt theo chiều rộng (BFS)

- Sử dụng một cấu trúc dữ liệu hàng đợi để lưu trữ thông tin trung gian thu được trong quá trình tìm kiếm:
- Chèn đỉnh gốc vào hàng đợi (đang hướng tới)
- Lấy ra đỉnh đầu tiên trong hàng đợi và quan sát nó
 - Nếu đỉnh này chính là đỉnh đích, dừng quá trình tìm kiếm và trả về kết quả.
 - Nếu không phải thì chèn tất cả các đỉnh kề với đỉnh vừa thăm nhưng chưa được quan sát trước đó vào hàng đợi.
- Nếu hàng đợi là rỗng, thì tất cả các đỉnh có thể đến được đều đã được quan sát – dừng việc tìm kiếm và trả về "không thấy".
- Nếu hàng đợi không rỗng thì quay về bước 2.



DUYỆT THEO CHIỀU RỘNG - BFS



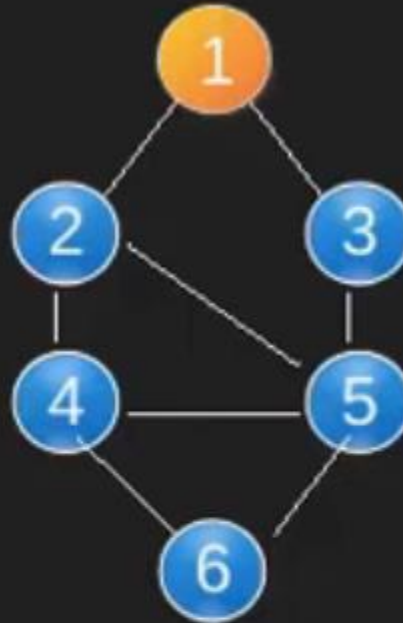
Visited :

1	2	3	4	5	6
0	0	0	0	0	0

Queue :



DUYỆT THEO CHIỀU RỘNG - BFS



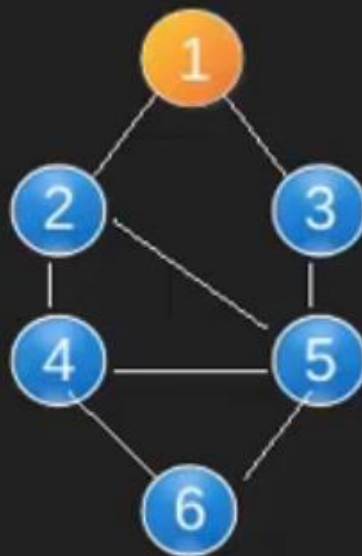
Visited :

1	2	3	4	5	6
1	0	0	0	0	0

Queue : 1



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

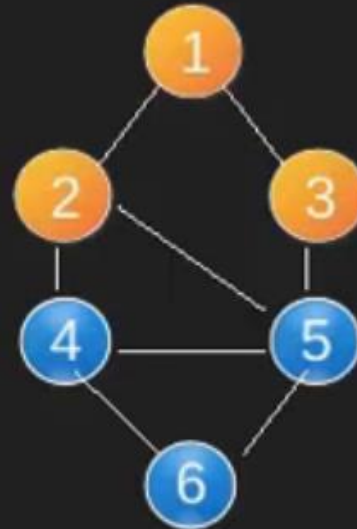
1	2	3	4	5	6
1	0	0	0	0	0

Queue :

Print : 1



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

1	2	3	4	5	6
1	1	1	0	0	0

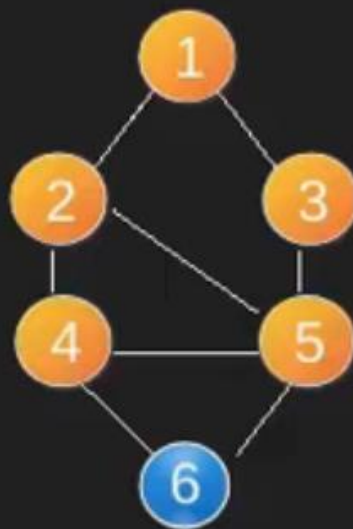
Queue : 2 3

After removing 1 from queue and printing it, we enqueue its non-visited adjacentNodes

Print : 1



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

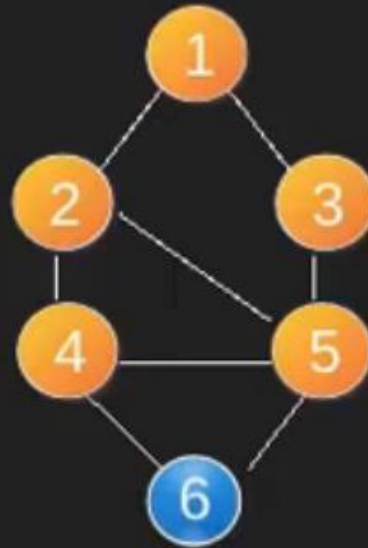
1	2	3	4	5	6
1	1	1	1	1	0

Queue : 3 4 5

Print : 1 2



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

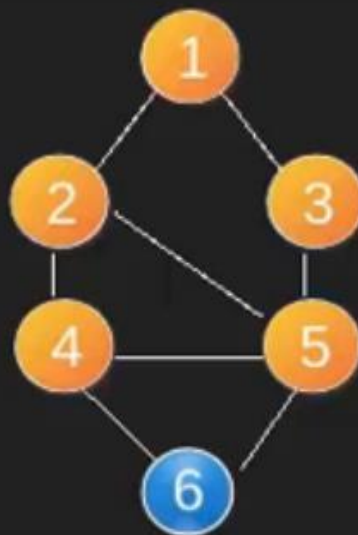
1	2	3	4	5	6
1	1	1	1	1	0

Queue : 4 5

Print : 1 2 3



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

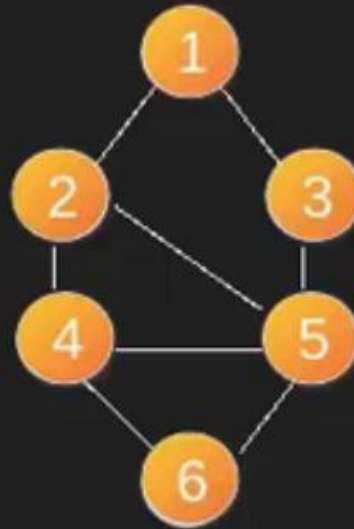
1	2	3	4	5	6
1	1	1	1	1	0

Queue : 5

Print : 1 2 3 4



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

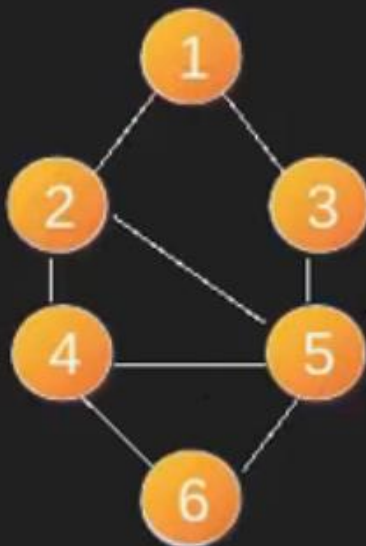
1	2	3	4	5	6
1	1	1	1	1	1

Queue : 5 6

Print : 1 2 3 4



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

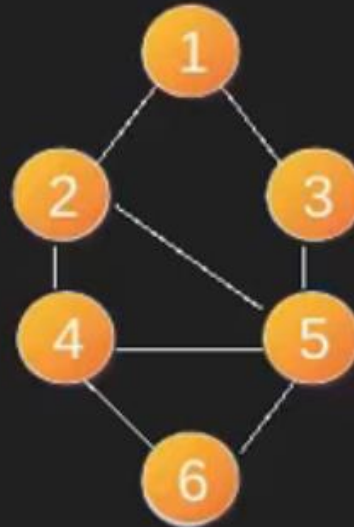
1	2	3	4	5	6
1	1	1	1	1	1

Queue : 6

Print : 1 2 3 4 5



DUYỆT THEO CHIỀU RỘNG - BFS



Visited :

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

Queue :

Print : 1 2 3 4 5 6



BÀI TẬP

1. Tìm hiểu và cài đặt hàm biểu diễn đồ thị bằng danh sách liên kết.
2. Viết hàm cài đặt xác định số thành phần liên thông của đồ thị và cho biết đồ thị có liên thông
3. Viết hàm cho biết đồ thị có chu trình không ? Nếu có liệt kê các chu trình của đồ thị?