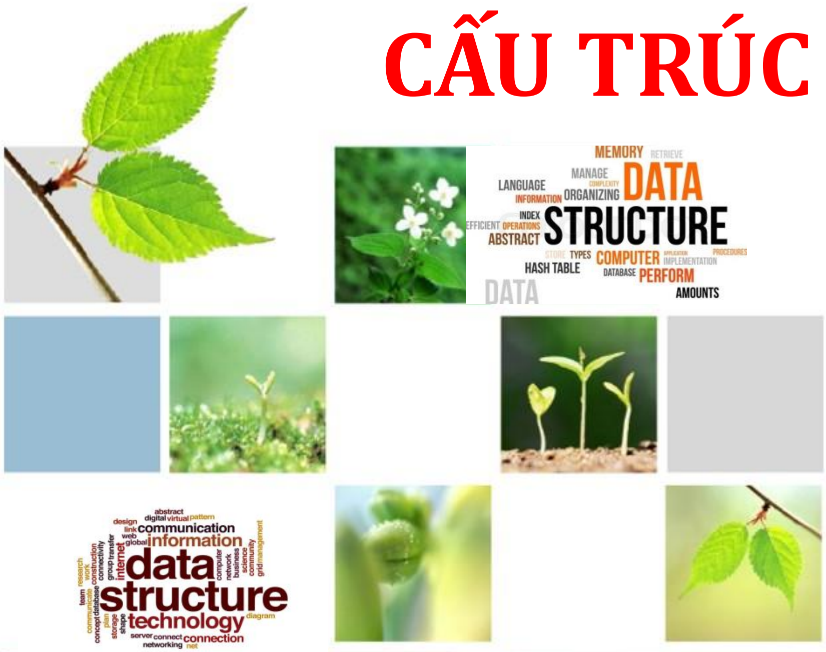
BUỔI 12: ĐỒ THỊ

LÝ THUYẾT

CẤU TRÚC DỮ LIỆU & GIẢI THUẬT

IT003.N210



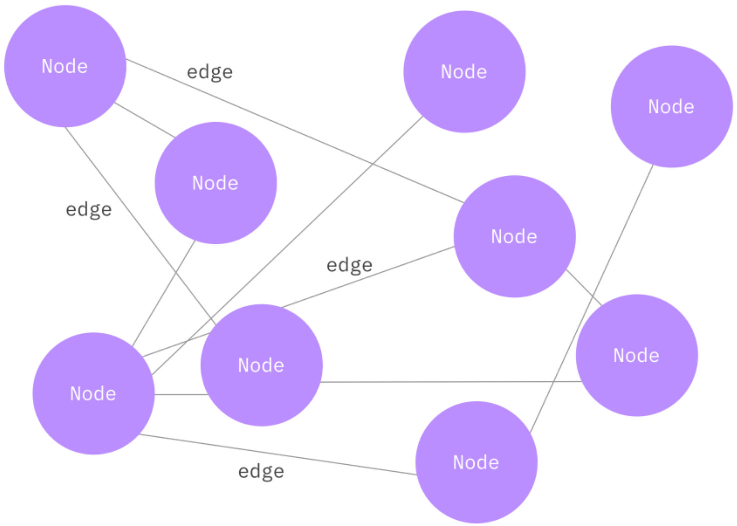
Giảng viên: Ths Đặng Văn Em

Email: vanem@uit.edu.vn

Điện thoại: 0966661006

BUỔI 12: ĐỒ THỊ

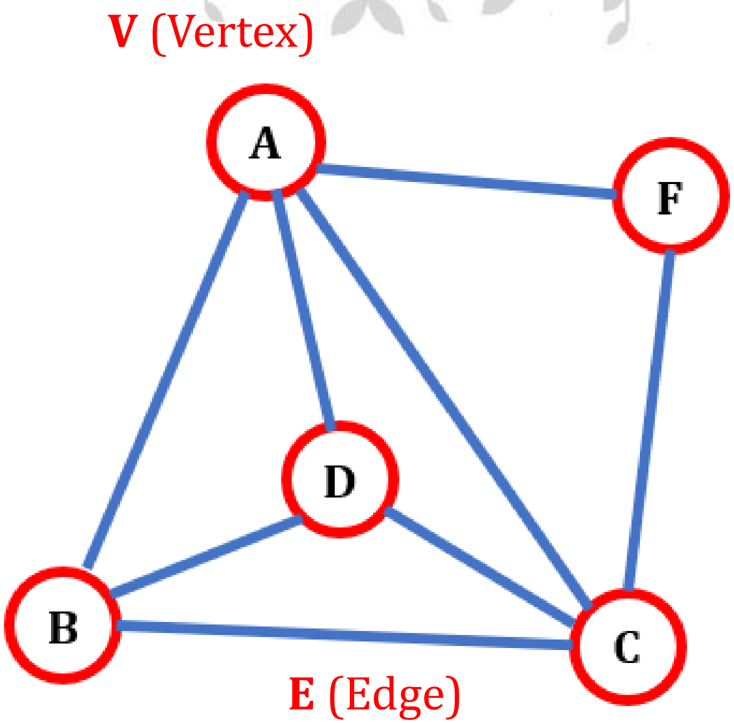
1) Đồ thị, các Khái niệm về Đồ thị.
2) Biểu diễn Đồ thị trên máy tính.
3) Các thuật toán duyệt Đồ thị (DFS, BFS)
4) Một số ứng dụng của tìm kiếm trên Đồ thị
5) Bài tập chương.



ĐỒ THỊ VÔ HƯỚNG ?

Một đồ thị **vô hướng** $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ được định nghĩa bởi:

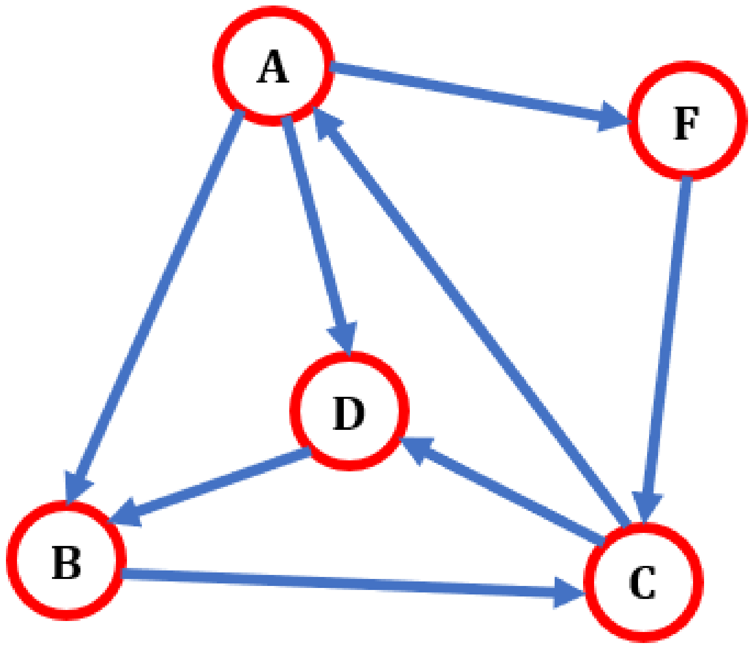
- Tập hợp **V** được gọi là tập các đỉnh;
- Tập hợp **E** là tập các cạnh;
- Mỗi cạnh $e \in E$ được liên kết với một cặp đỉnh $\{i, j\} \in V^2$, không phân biệt thứ tự



ĐỒ THỊ CÓ HƯỚNG ?

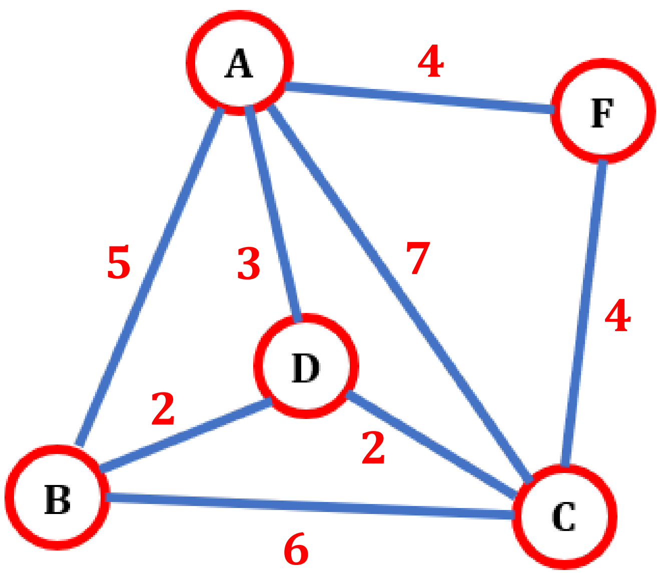
Một đồ thị **có hướng** $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ được định nghĩa bởi:

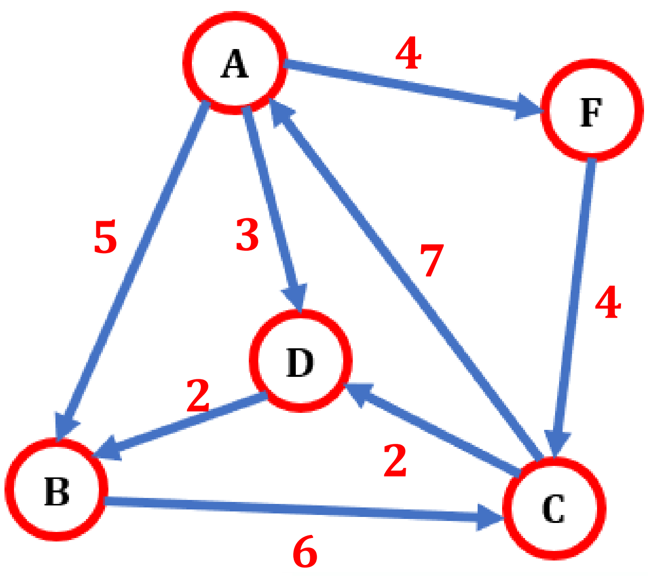
- ❖ Tập hợp **V** được gọi là tập các đỉnh;
- ❖ Tập hợp **U** là tập các cạnh;
- ❖ Mỗi cạnh $u \in U$ được liên kết với một cặp đỉnh $(i, j) \in \mathrm{V}^2$



ĐỒ THỊ CÓ TRỌNG SỐ?

Một đồ thị $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ được gọi là có trọng số nếu mỗi cạnh của G có gắn một con số để biểu diễn thuộc tính nào đó của cạnh:

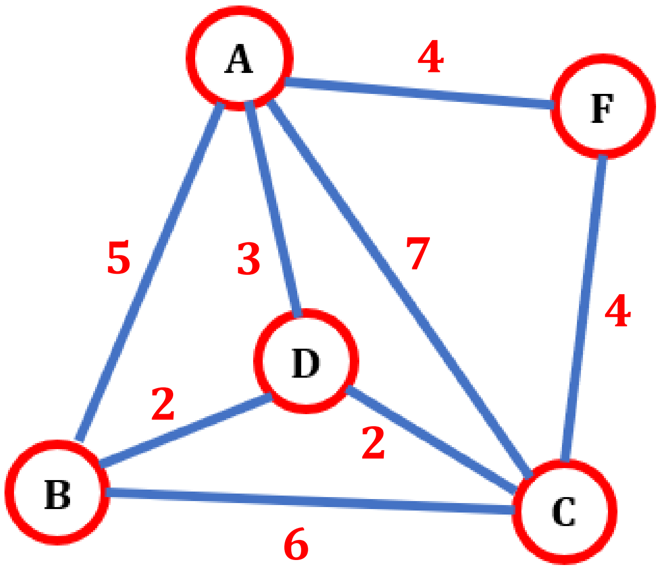


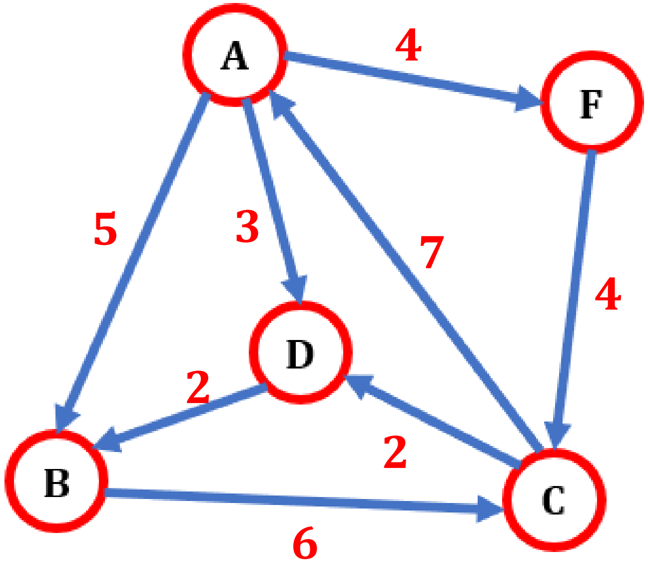


BẬC CỦA ĐỈNH

❖ Bậc của đỉnh trên đồ thị vô hướng: **Deg (v)**

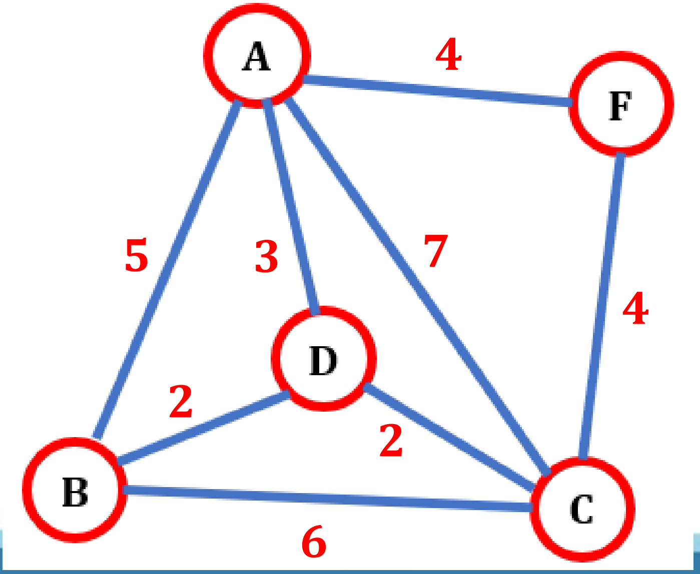
❖ Bậc của đỉnh trên đồ thi có hướng: **Deg (v) = Deg +(v) + Deg -(v)**

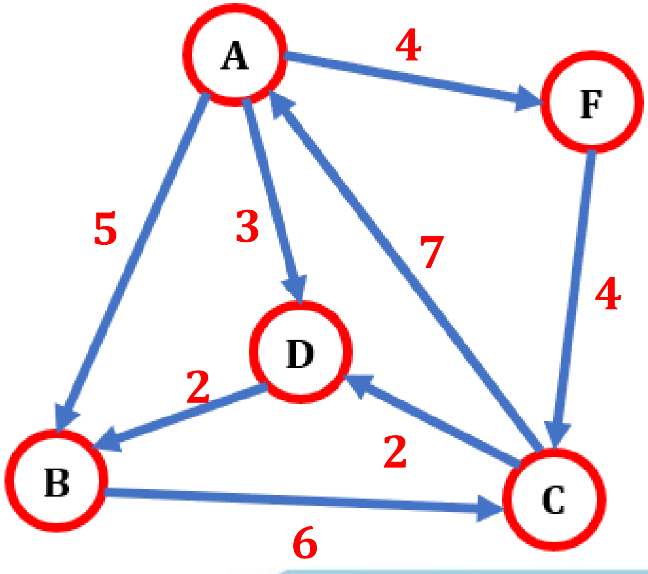




BẬC CỦA ĐỒ THỊ

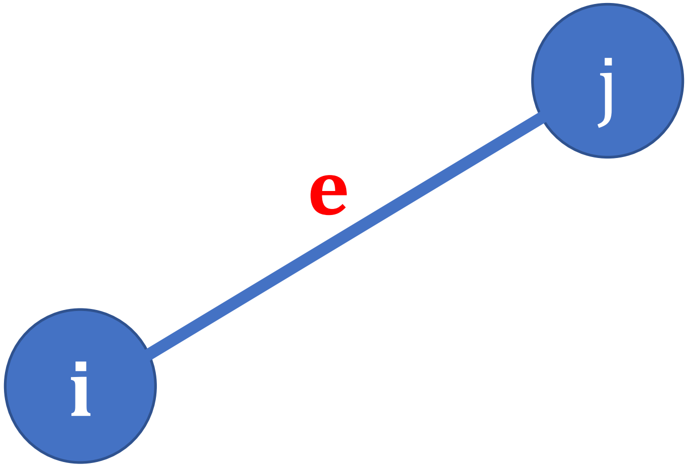
- Bậc của đồ thị là tổng bậc của mọi đỉnh trên đồ thị: $\sum \mathrm{Deg}(v)$
- Định lý tổng bậc: $\sum \boldsymbol{Deg(v)} = \mathbf{2.E}$
- Số đỉnh bậc lẻ luôn luôn là số chẵn





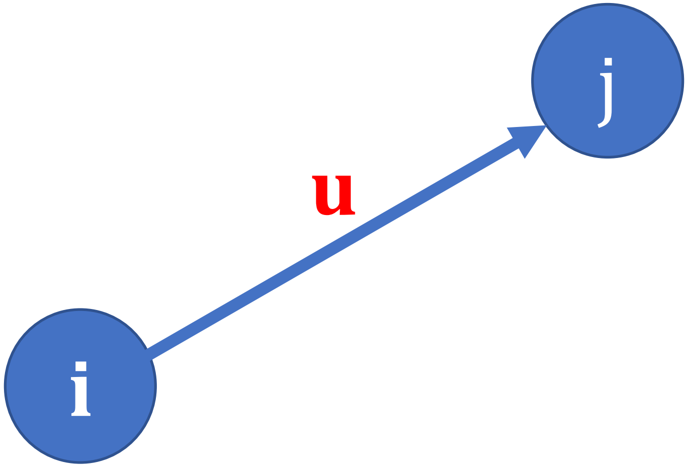
ĐỈNH KỀ?

- 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).

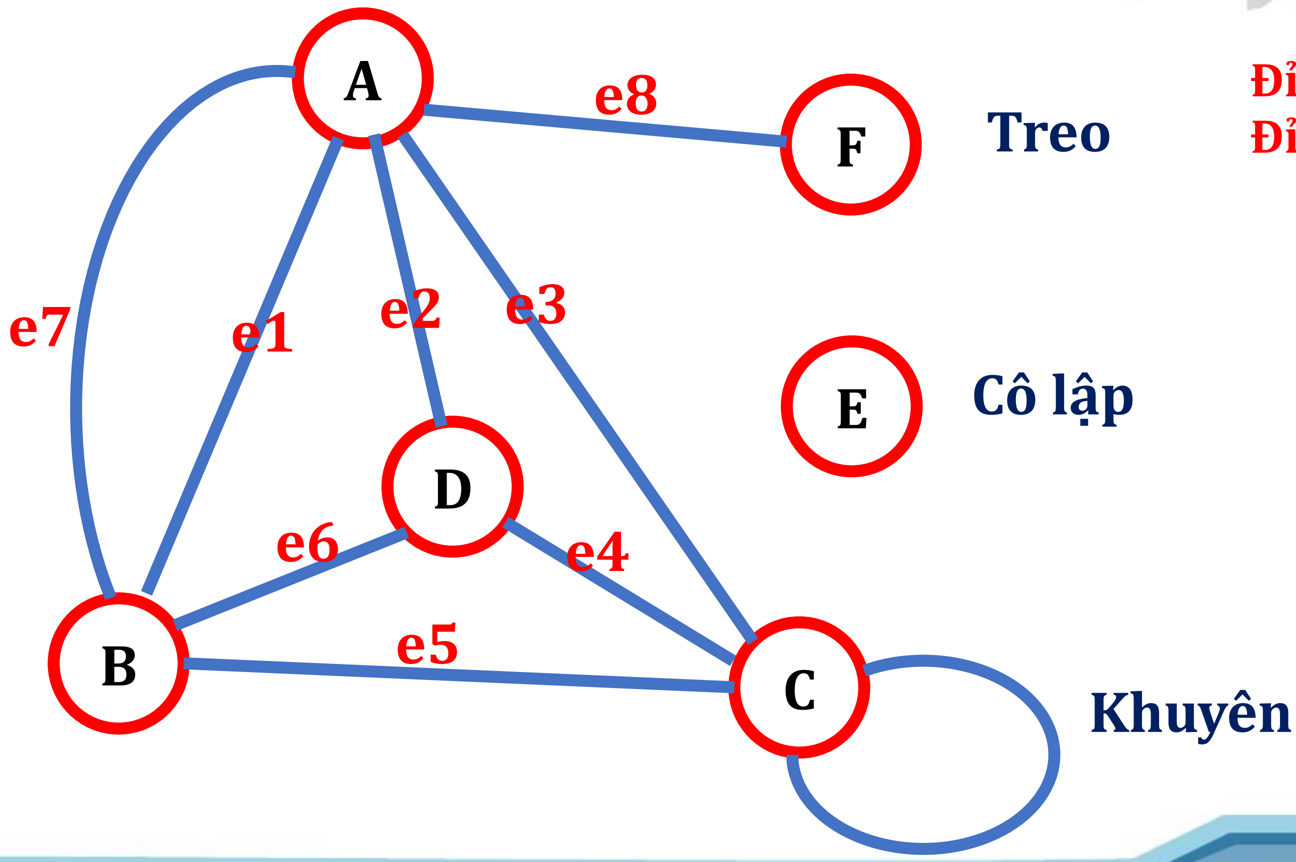


ĐỈNH KỀ?

- 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



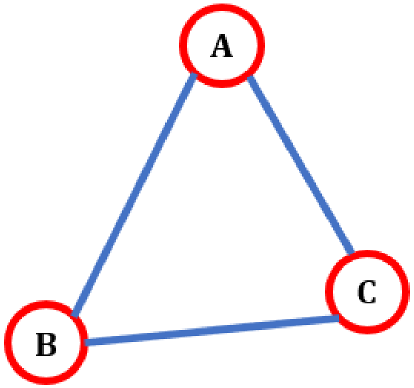
ĐỈNH TREO, ĐỈNH CÔ LẬP, KHUYÊN?



Đỉnh Treo BẬC = 1
Đỉnh Cô lập BẬC=0

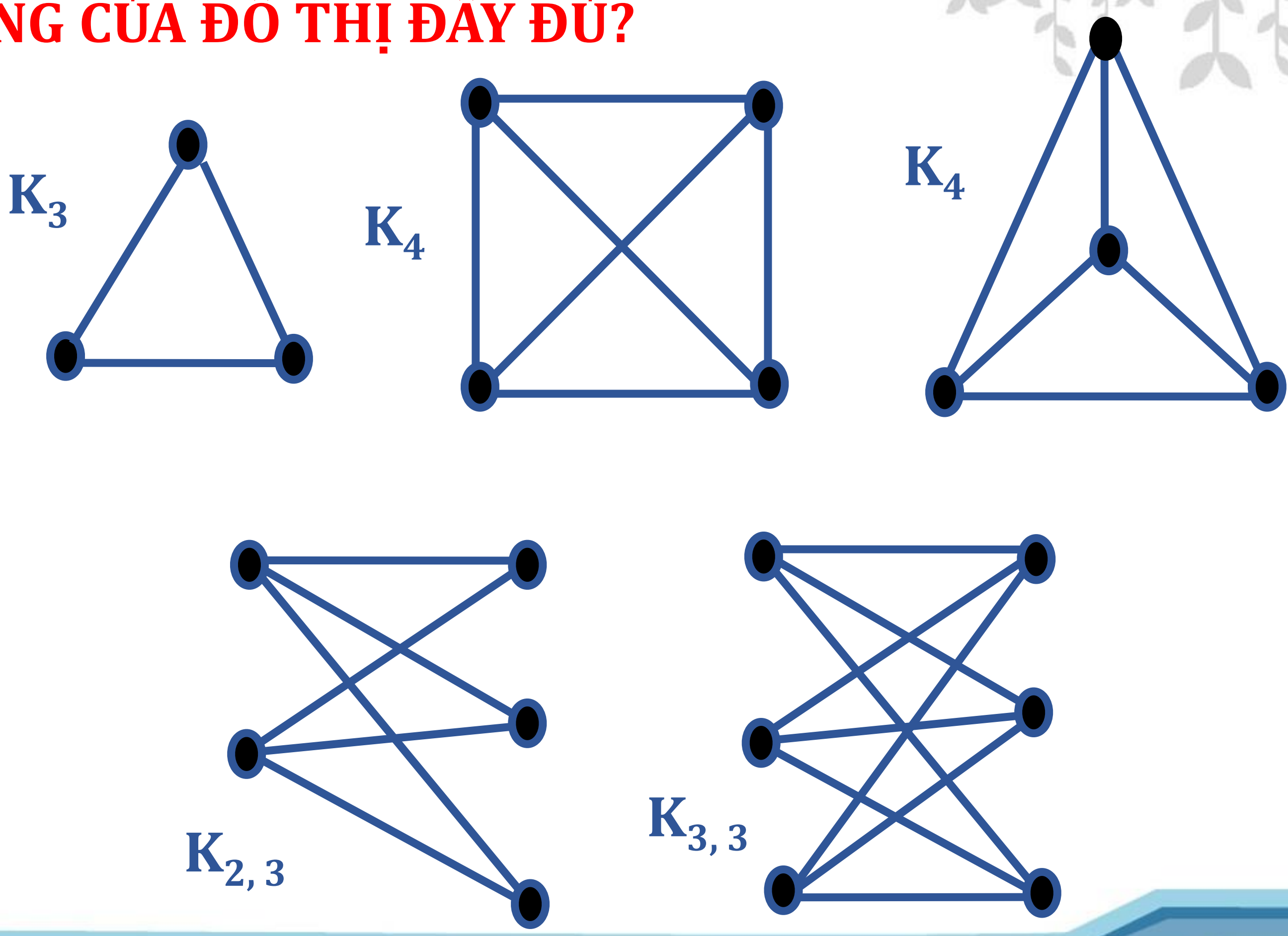
CÁC DẠNG CỦA ĐỒ THỊ?

- Đồ thị **RỖNG**: tập cạnh là tập rỗng
- Đồ thị **ĐƠN**: không có khuyên và cạnh song song.
- Đồ thị **PHẲNG**: không có 2 cạnh cắt nhau.
- 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



ĐỒ THỊ, CÁC KHÁI NIỆM ĐỒ THỊ

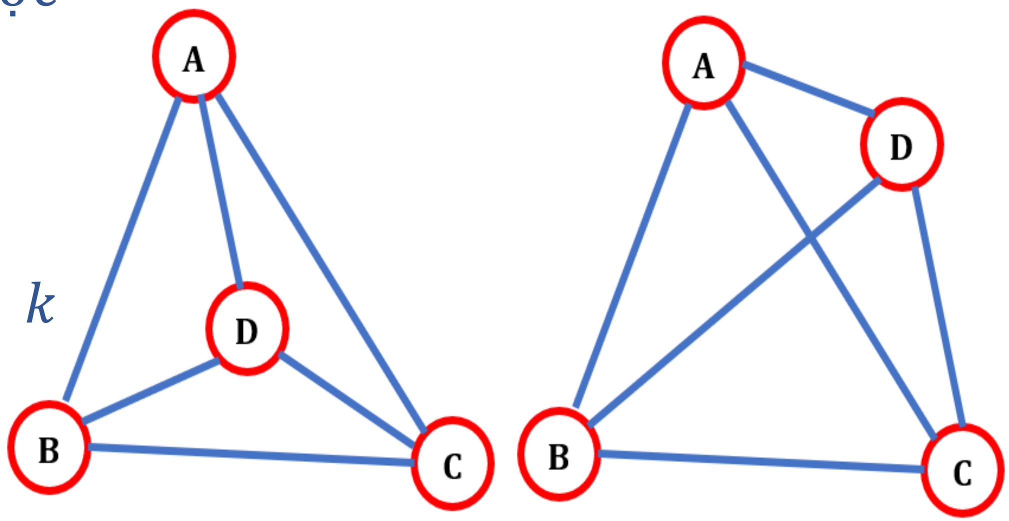
CÁC DẠNG CỦA ĐỒ THỊ ĐẦY ĐỦ?



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

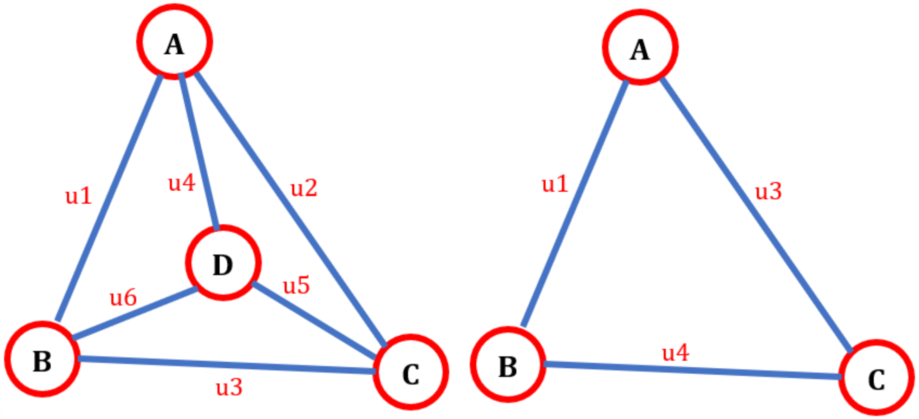


ĐỒ THỊ, CÁC KHÁI NIỆM ĐỒ THỊ

ĐỒ THỊ CON?

Xét hai đồ thị $\mathbf{G} = (\mathbf{X}, \mathbf{U})$ và $\mathbf{G_1} = (\mathbf{X_1}, \mathbf{U_1})$. 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 $i, j \in X_1$



BÀI TẬP?

Câu 1: Vẽ đồ thị có 6 đỉnh trong đó:

a) 3 đỉnh bậc 3 và 3 đỉnh bậc 1

b) Các đỉnh có bậc là 1,2,2,3,4,5

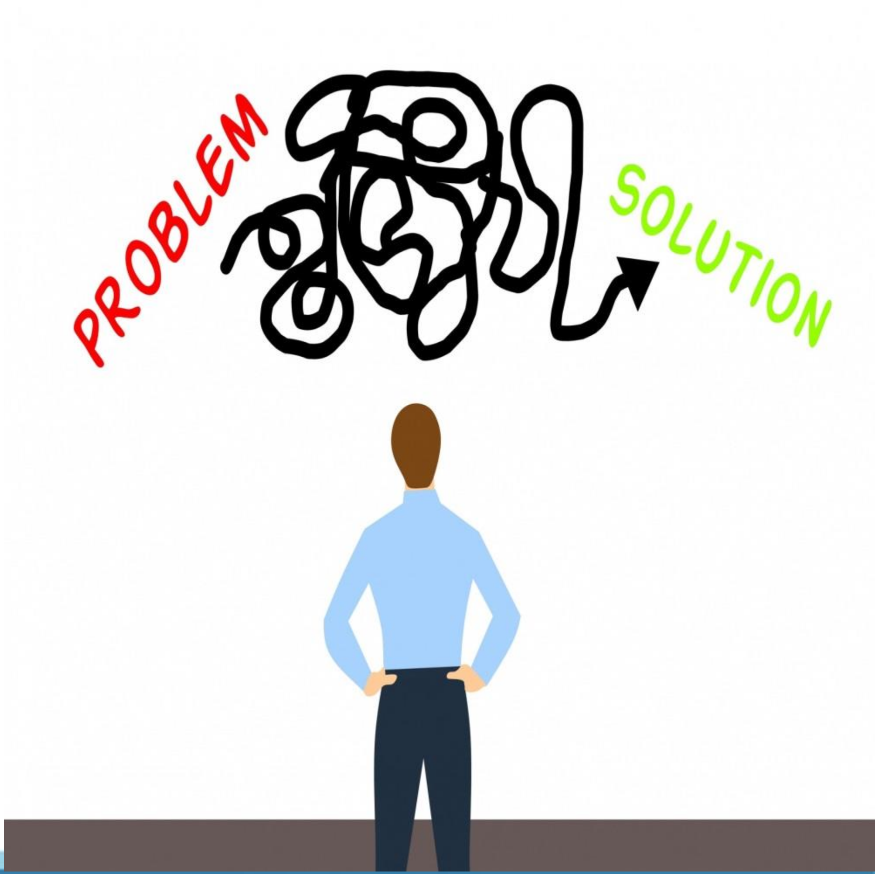
c) Bậc các đỉnh 2,2,4,4,4,4

Câu 2: Vẽ đồ thị mọi đỉnh bậc 3 có:

a) Có 4 đỉnh

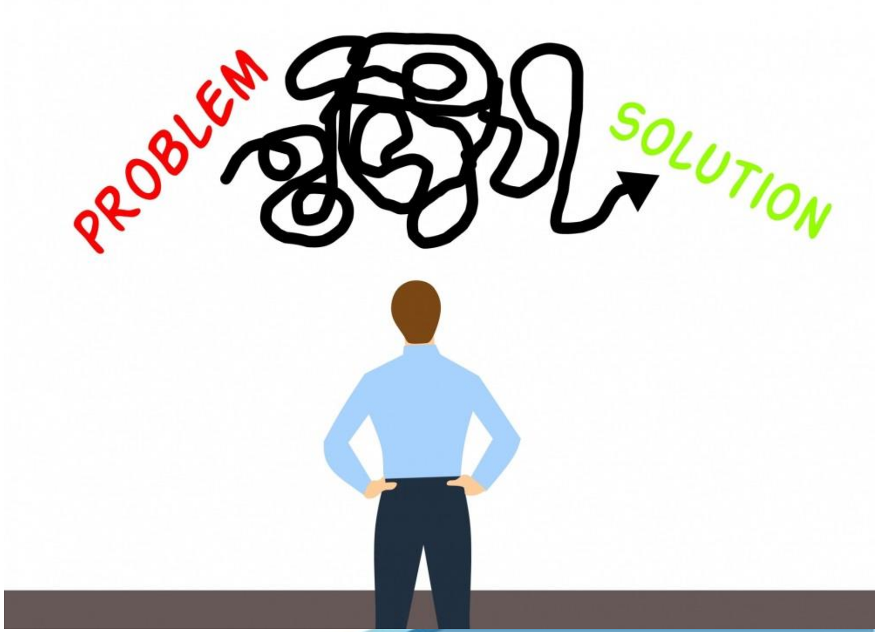
b) Có 5 đỉnh

c) Có 6 đỉnh



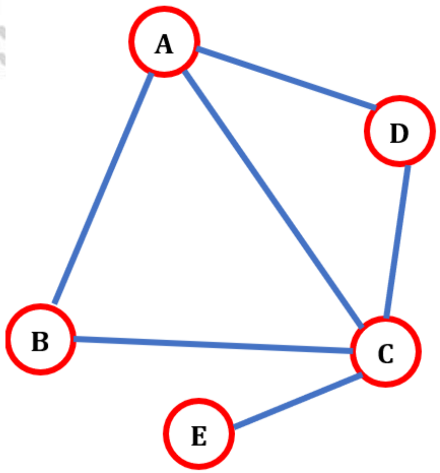
BÀI TẬP?

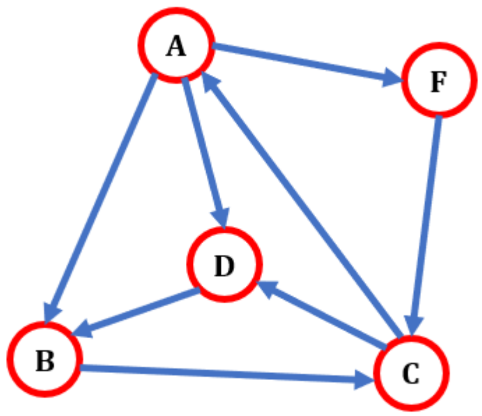
Câu 3: Một đồ thi có 19 cạnh, mỗi đỉnh đều có bậc >=3. Hỏi đồ thị này có tối đa bao nhiêu đỉnh ?



ĐƯỜNG ĐI – CHU TRÌNH

- **Đường đi (Path):** Cho đồ thị G(V,E) một đường đi độ dài n trên G là một dãy liên tiếp (n+1) đỉnh.
- Tập v1, v2, v3, v4…vn. Sao cho (vi, vi+1) là một cạnh (cung) của G. Đường đi mà không có cạnh nào lặp lại thì gọi là đường đi đơn.





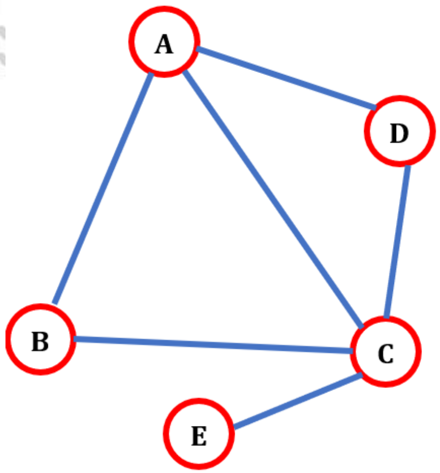
Đường đi đơn ?? Không là đường đi ?

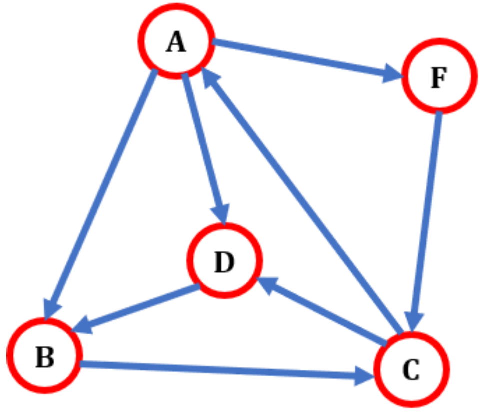
ĐƯỜNG ĐI – CHU TRÌNH

❖ **Chu trình (Cycle):** Là đường đi mà đỉnh xuất phát trùng đỉnh kết thúc. Chu trình mà không có cạnh nào đi qua 2 lần gọi chu trình đơn.

Chu trình đơn ??

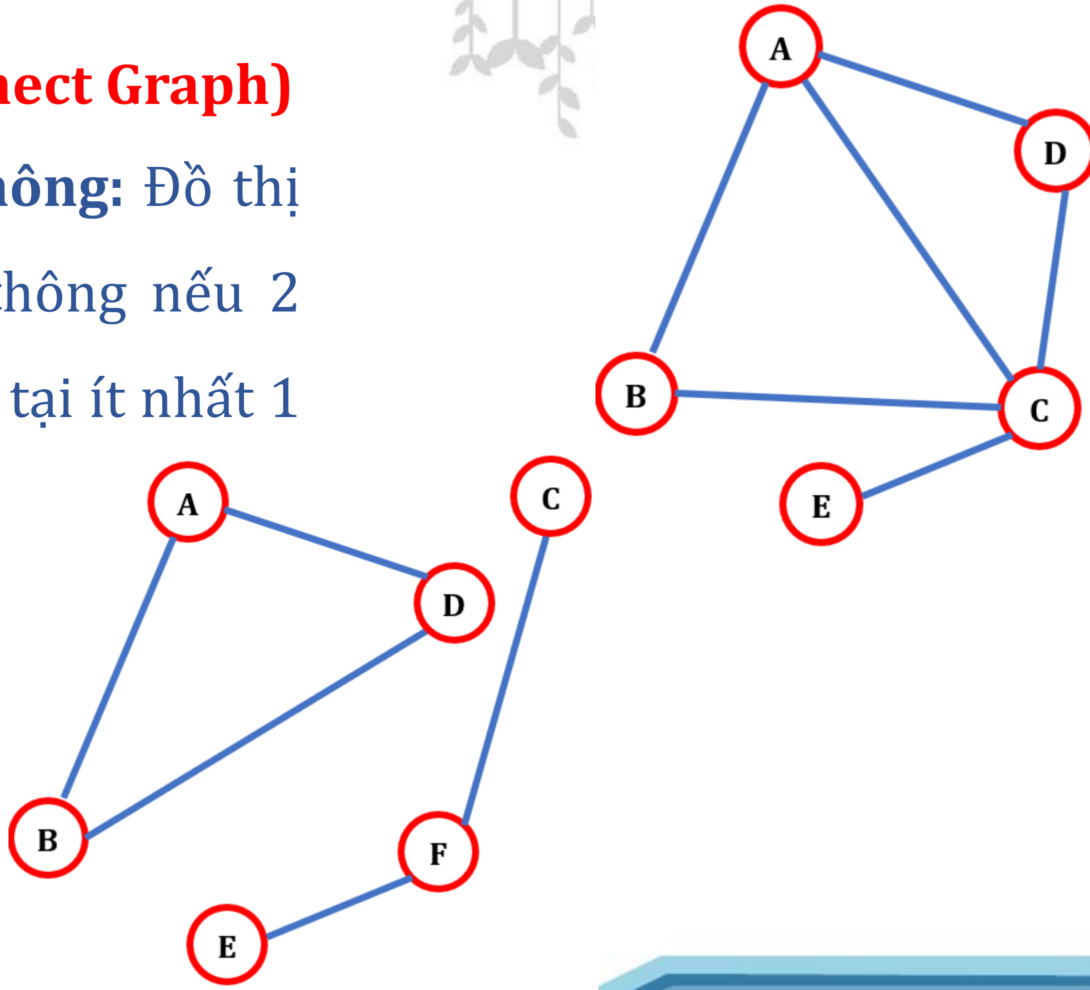
Chu trình không đơn??





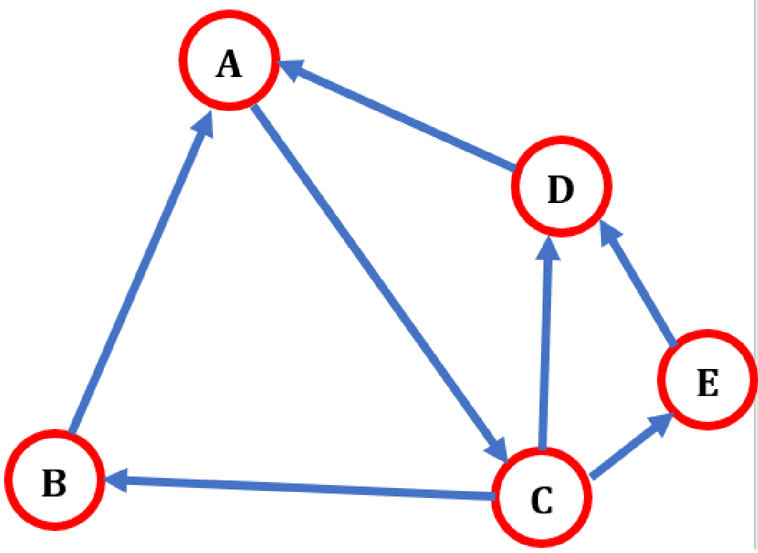
ĐỒ THỊ LIÊN THÔNG (Connect Graph)

- **Đồ thị vô hướng liên thông:** Đồ thị vô hướng G gọi là liên thông nếu 2 đỉnh tùy ý của G luôn tồn tại ít nhất 1 đường nối chúng.



ĐỒ THỊ LIÊN THÔNG (Connect Graph)

- **Đồ thị có hướng liên thông mạnh:** Nếu giữa 2 điểm bất kỳ luôn tồn tại ít nhất 1 con đường đi
- **Đồ thi có hướng liên thông yếu:** Nếu bỏ hướng các cung trên đồ thị thì ta được 1 đồ thi vô hướng liên thông.



Vẽ đồ thị liên thông mạnh ?

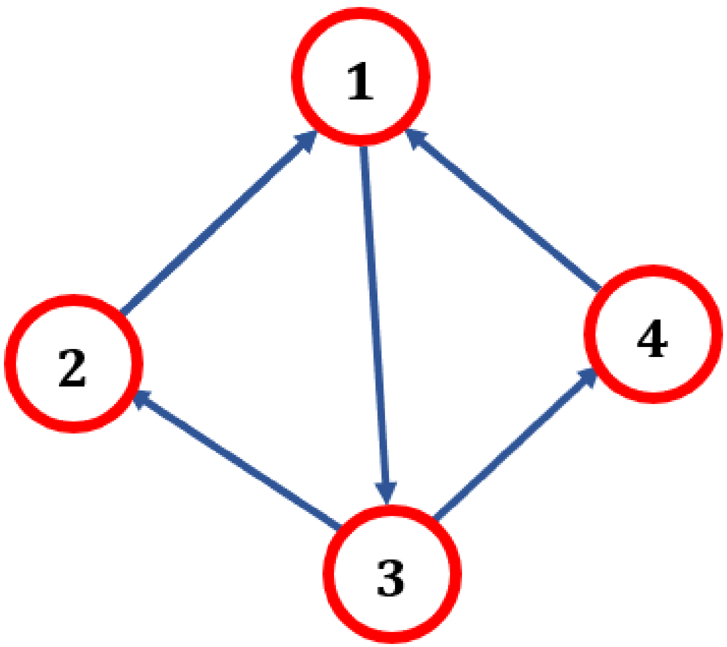
Vẽ đồ thị liên thông yếu?

BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ

Ma trận KỀ đồ thị có hướng:

- Xét đồ thị $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, giả sử tập **V** gồm n đỉnh và được sắp thứ tự $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_\mathrm{N}\}$, tập **E** gồm m cạnh và được sắp thứ tự $\mathrm{E} = \{\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_\mathrm{M}\}$.
- Ma trận kề của đồ thị G, ký hiệu $B(G)$, là một ma trận nhị phân cấp NxN: $B = (B_{ij})$ với B_{ij} được định nghĩa:
  - $\mathbf{B_{i,j} = 1}$ nếu có cạnh nối $\mathrm{x_i}$ tới $\mathrm{x_j}$,
  - $\mathbf{B_{i,j} = 0}$ trong trường hợp ngược lại

BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ



G

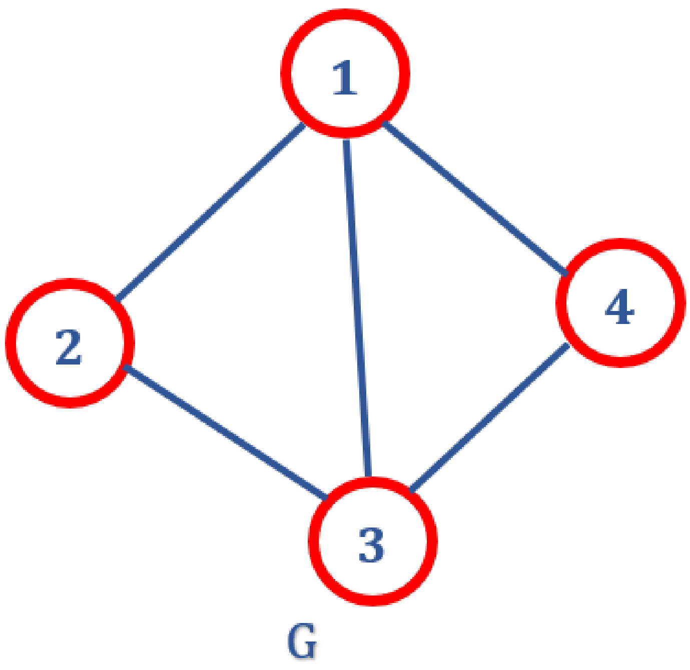
$$B = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ

Ma trận của đồ thị vô hướng:

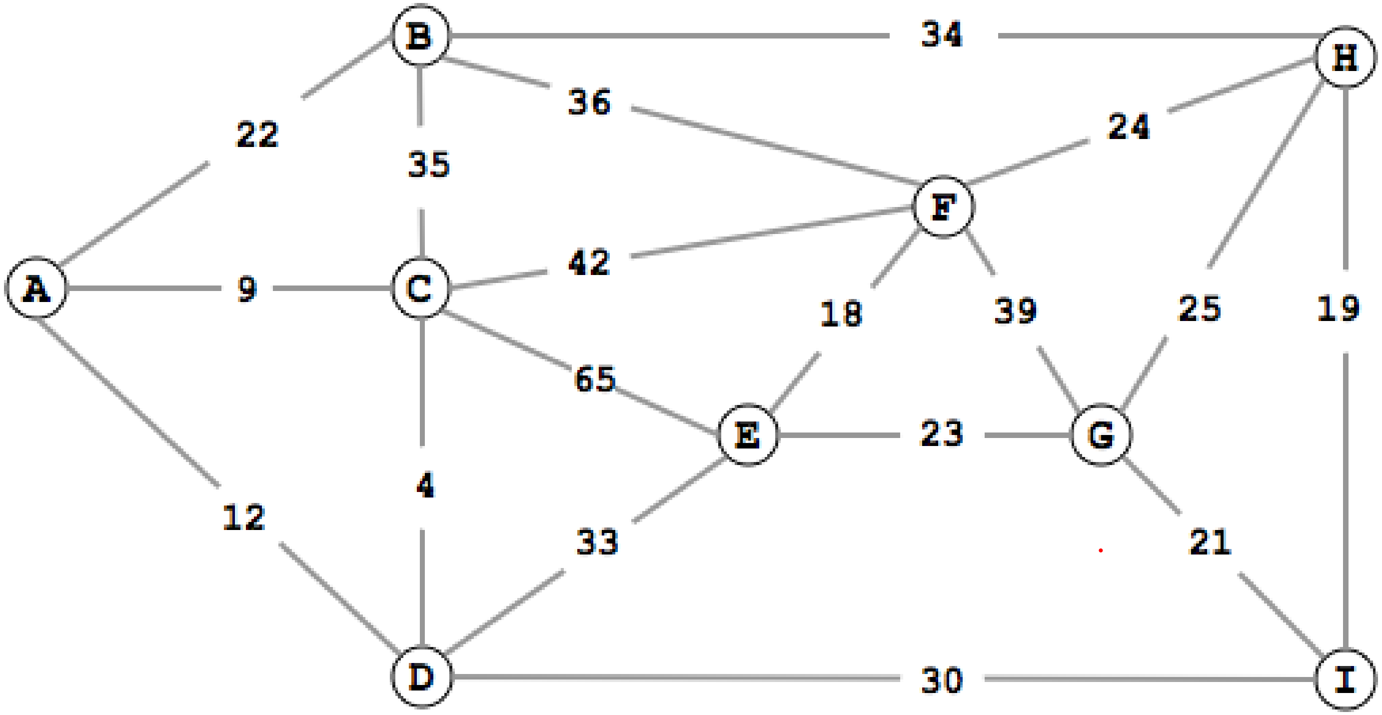
- Xét đồ thị $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ vô hướng, giả sử tập **V** gồm n đỉnh và được sắp thứ tự $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \dots, \mathrm{v}_\mathrm{N}\}$, tập **E** gồm m cạnh và được sắp thứ tự $\mathrm{E} = \{\mathrm{e}_1, \mathrm{e}_2, \dots, \mathrm{e}_\mathrm{M}\}$.
- Ma trận của G, ký hiệu $A(G)$, là ma trận nhị phân NxM: $A = (A_{ij})$ với A_{ij} được định nghĩa:
  - **$\mathbf{A_{ij} = 1}$ nếu đỉnh $\mathbf{x_i}$ kề với cạnh $\mathbf{u_j}$,**
  - **$\mathbf{A_{ij} = 1}$ nếu ngược lại**

BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ

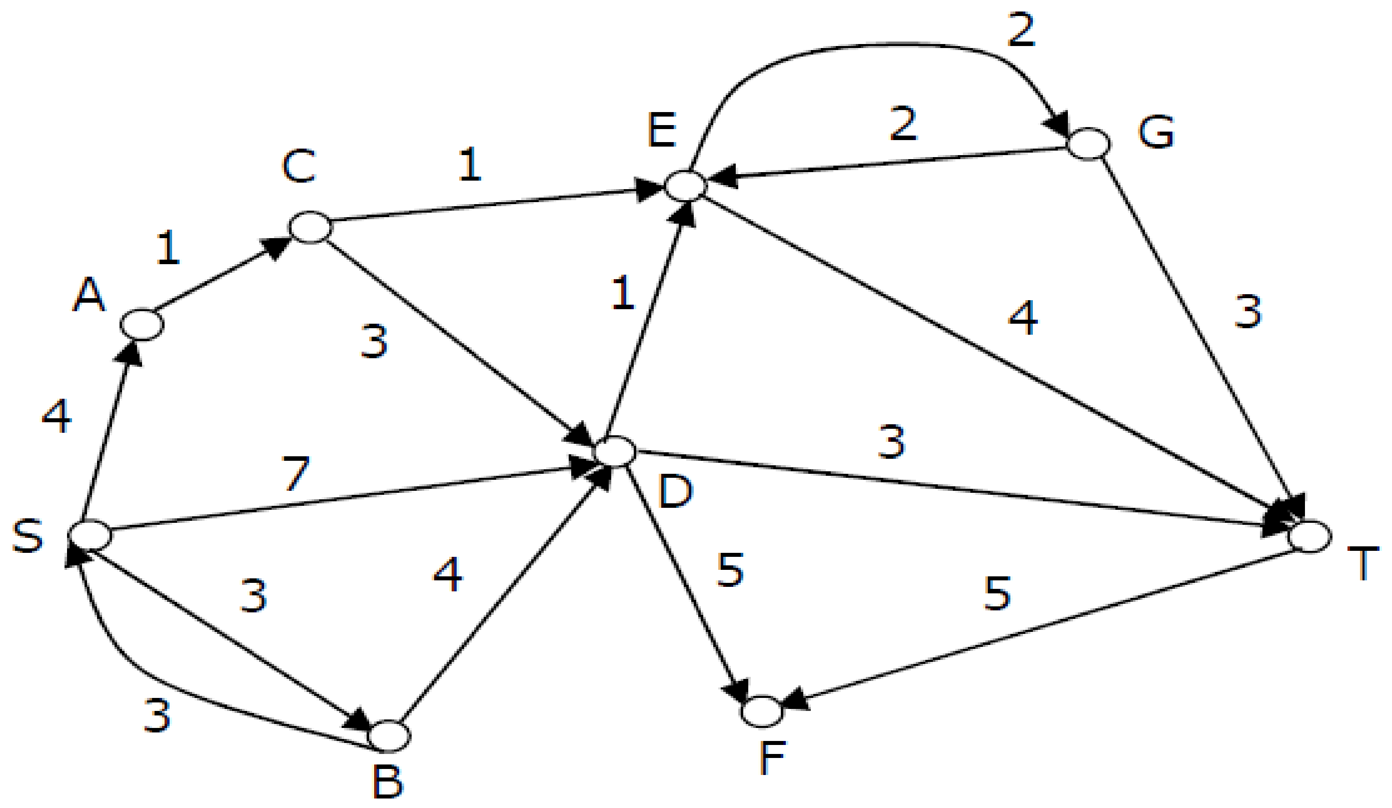


$$B = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ



BIỂU DIỄN ĐỒ THỊ BẰNG MA TRẬN KỀ



CÀI ĐẶT BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Khai báo

Nhập đồ thị

Xuất đồ thị

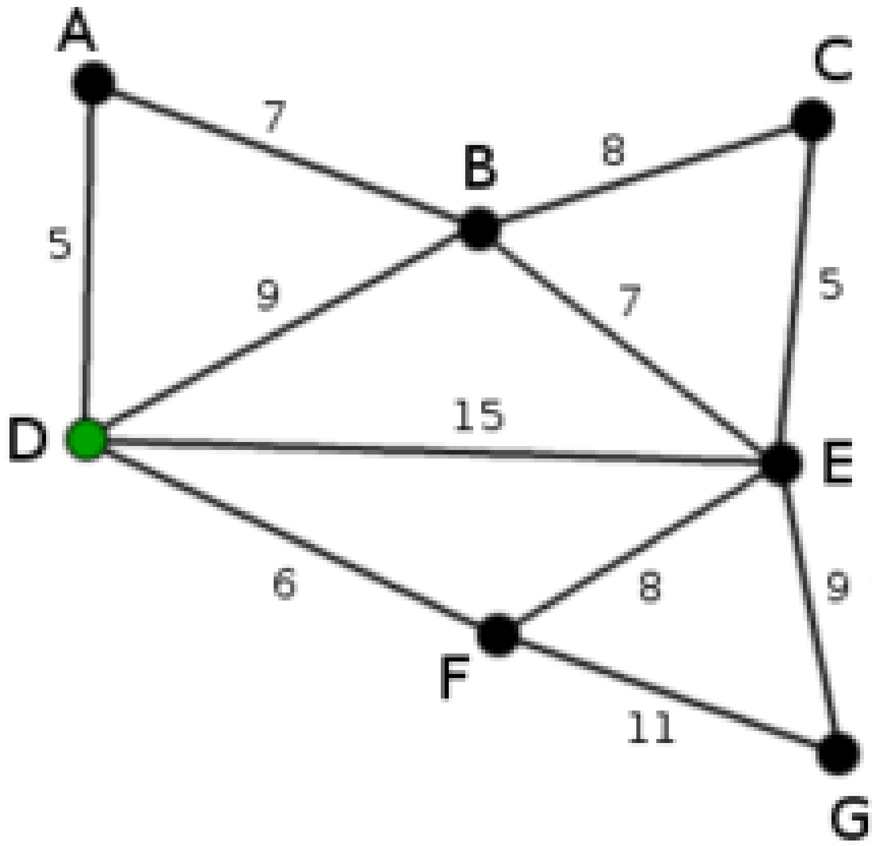
THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)

Cho $G = (V, E)$ là đồ thị có tập các đỉnh V và tập các cạnh E..

- ❖ Xuất phát từ 1 đỉnh tùy ý duyệt đỉnh này (gán nhãn v=1)và sau đó duyệt đỉnh kề nào đó.
- ❖ Ở bước tiếp theo ta duyệt đỉnh kề chưa được duyệt của đỉnh vừa duyệt. Khi duyệt một đỉnh ta đánh dấu là đã duyệt đỉnh đó.
- ❖ Tiếp tục như thế cho đến khi đỉnh vừa duyệt không còn đỉnh kề nào chưa duyệt (đi hết chiều sâu). Quay lại đỉnh đã duyệt ở bước kế trên. Nếu còn đỉnh nào chưa duyệt thì tiếp tục duyệt. Thực hiện đến khi không còn đỉnh nào chưa được duyệt.

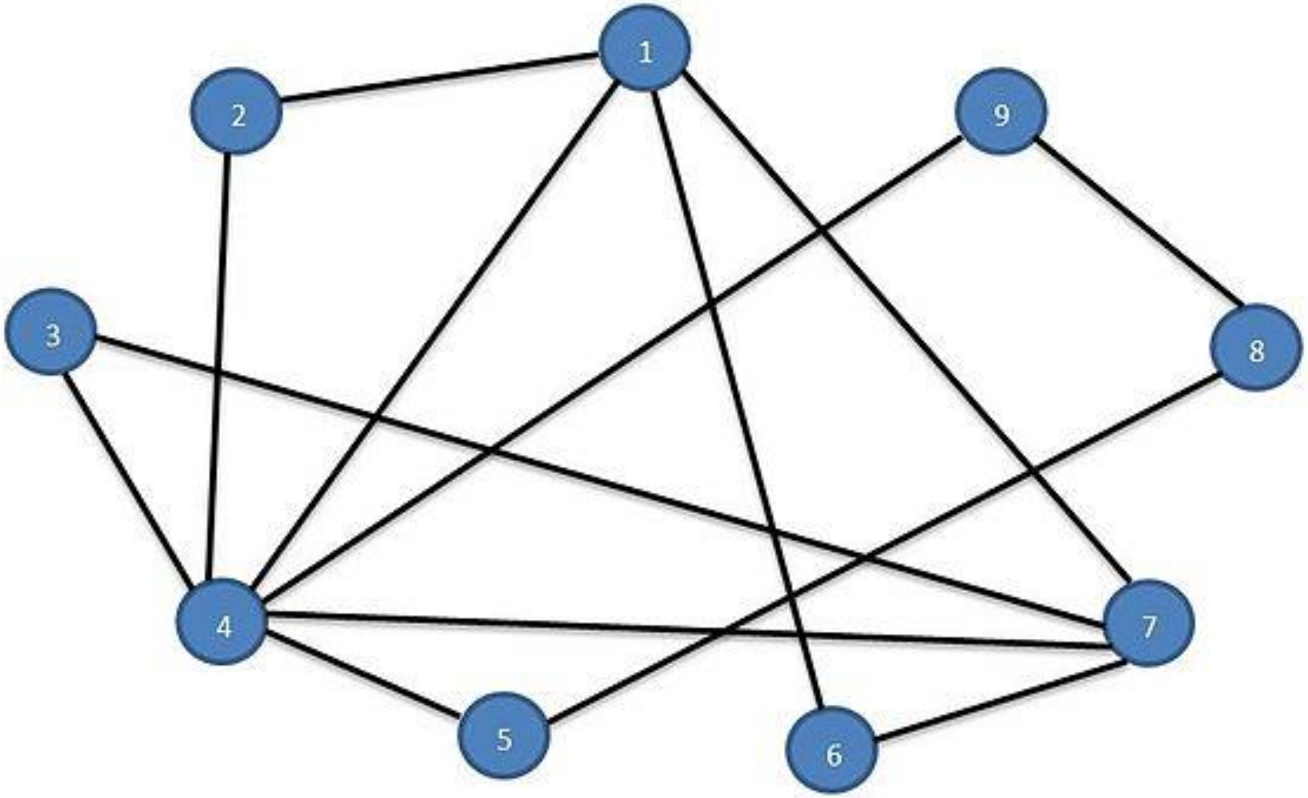
THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)

.



THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)

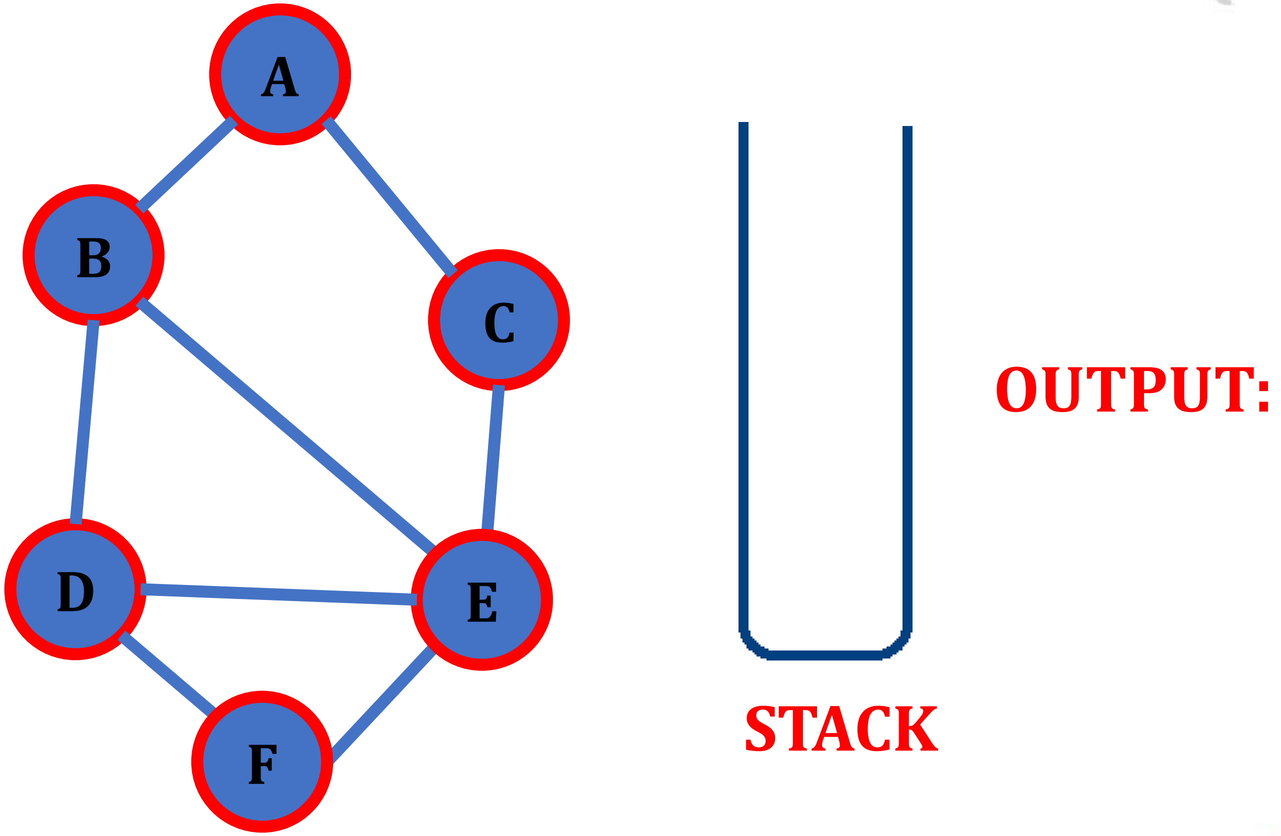
.



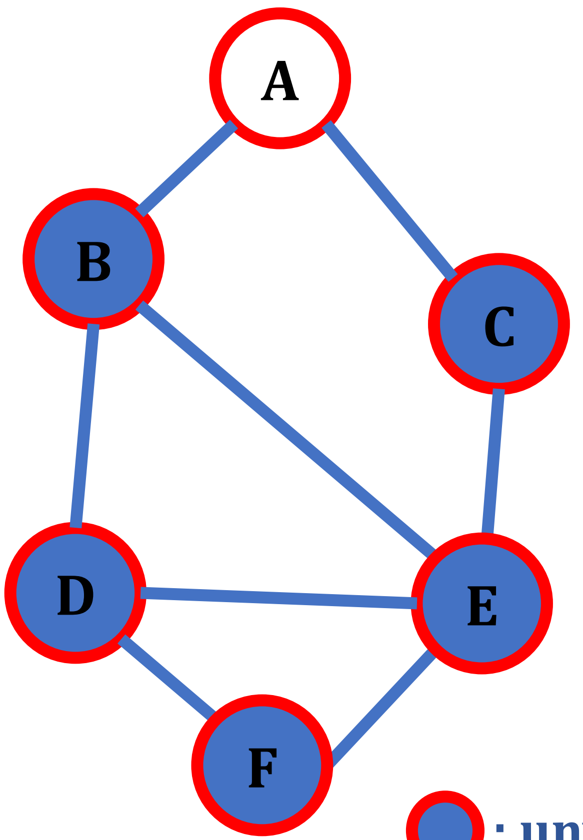
THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)

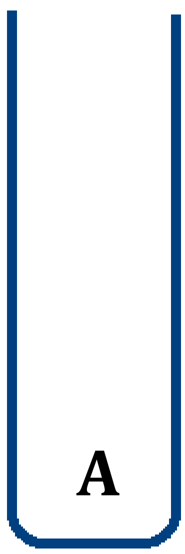
```
void DFS (int v)
{
   //Gắn nhãn v đã duyệt;
   for (u = 1; u <= n; u++)
    if (u tồn tại trong danh sách kề V)
      if(u có nhãn là 0)
         {
            Xử lý đỉnh u;  //Gắn nhãn 1
            DFS (u);
         }
}
```

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



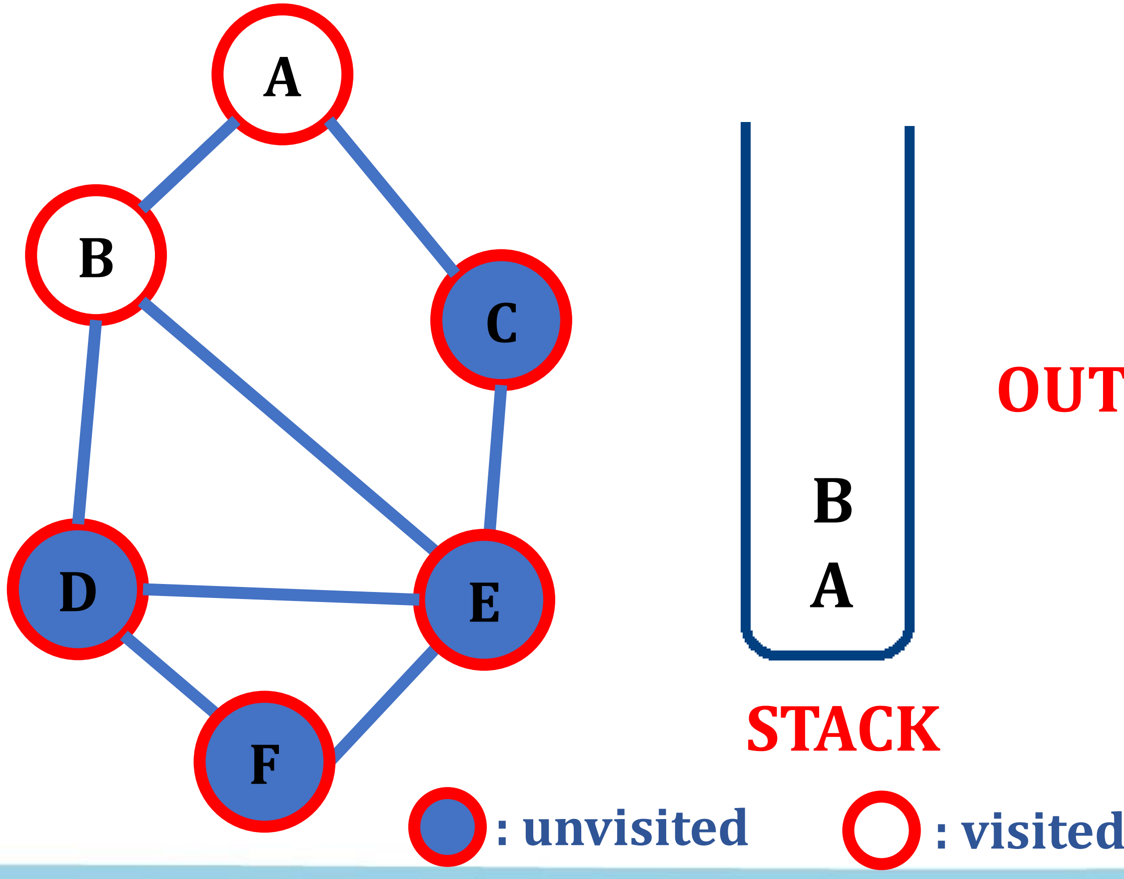


STACK

OUTPUT: A

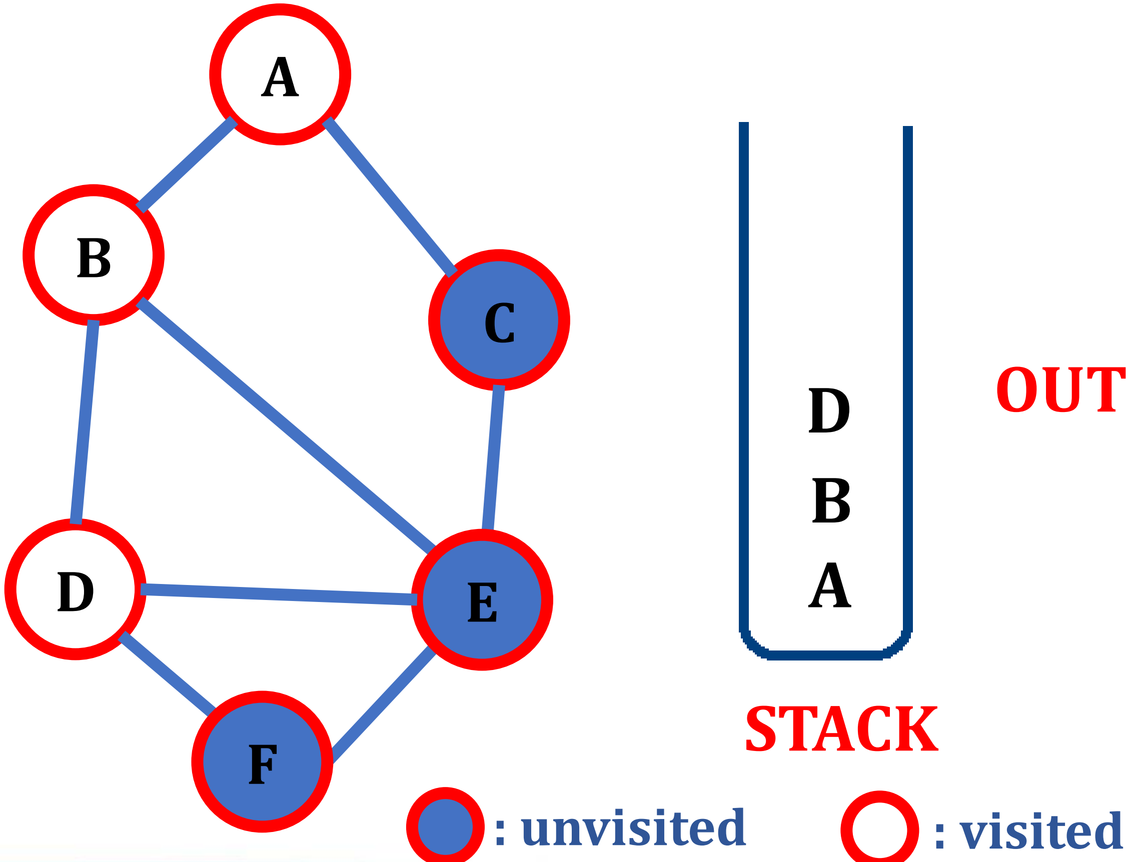
: unvisited   : visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



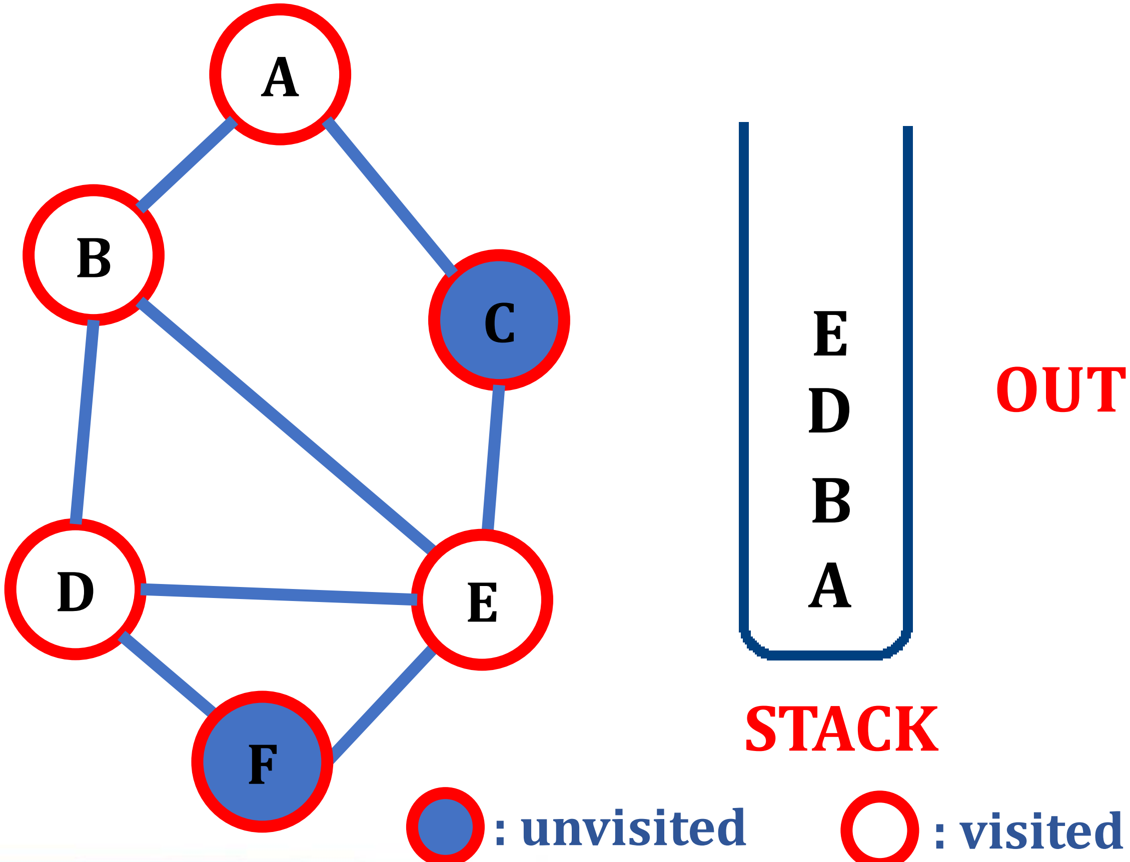
OUTPUT: A B

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



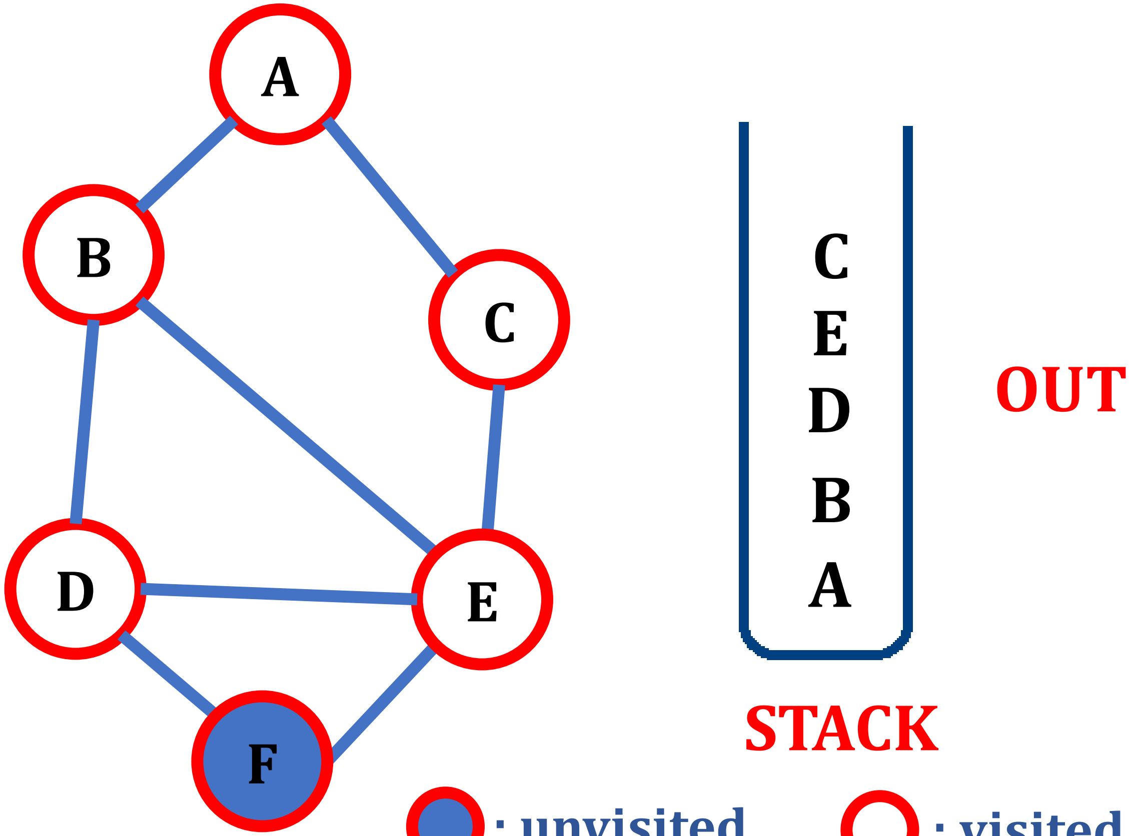
OUTPUT: A B D

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



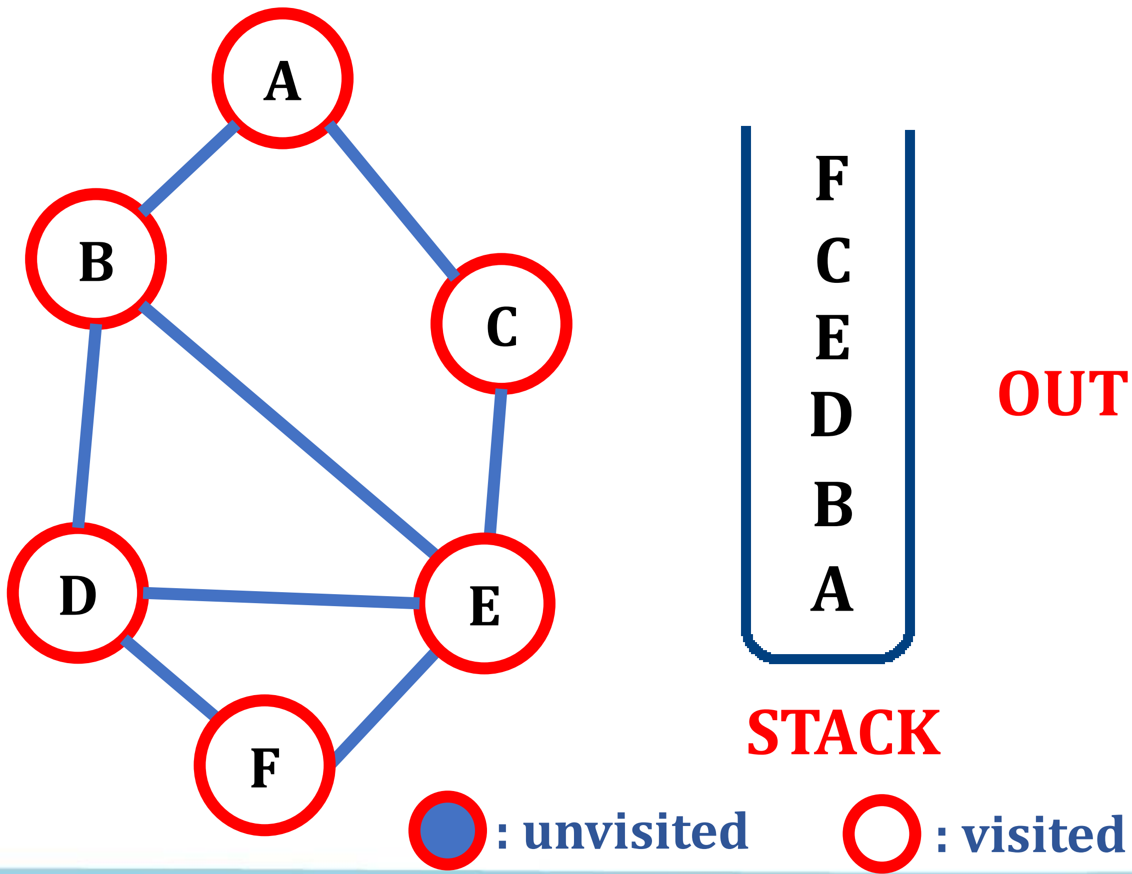
OUTPUT: A B D E

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



OUTPUT: A B D E C

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (DFS)



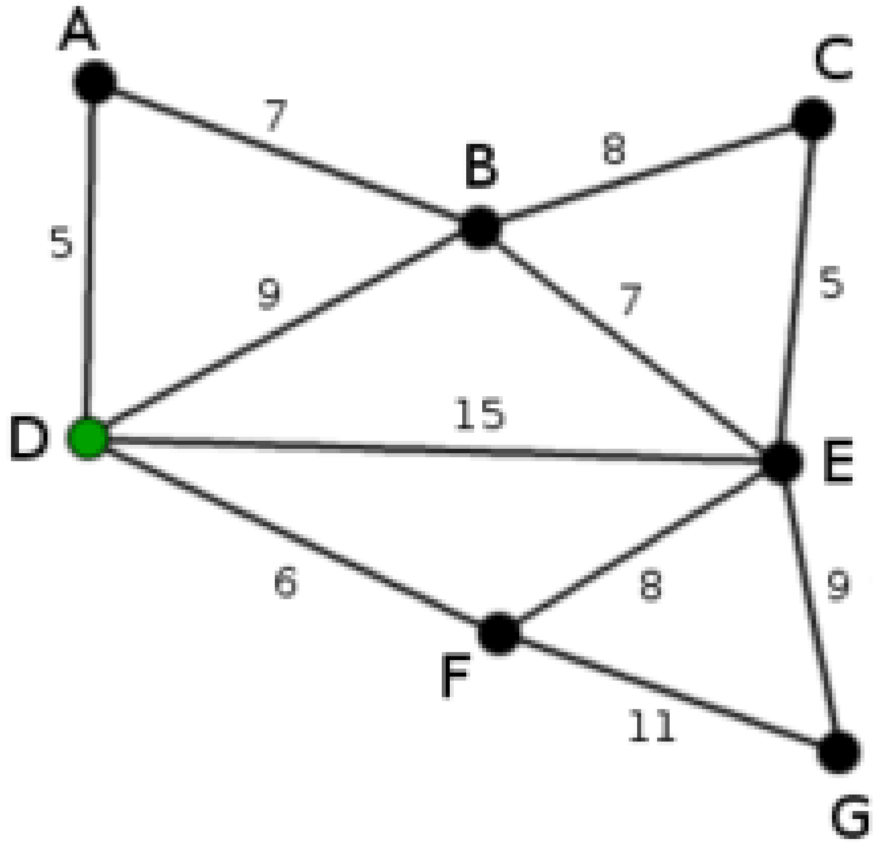
OUTPUT: A B D E C F

THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG (BFS)

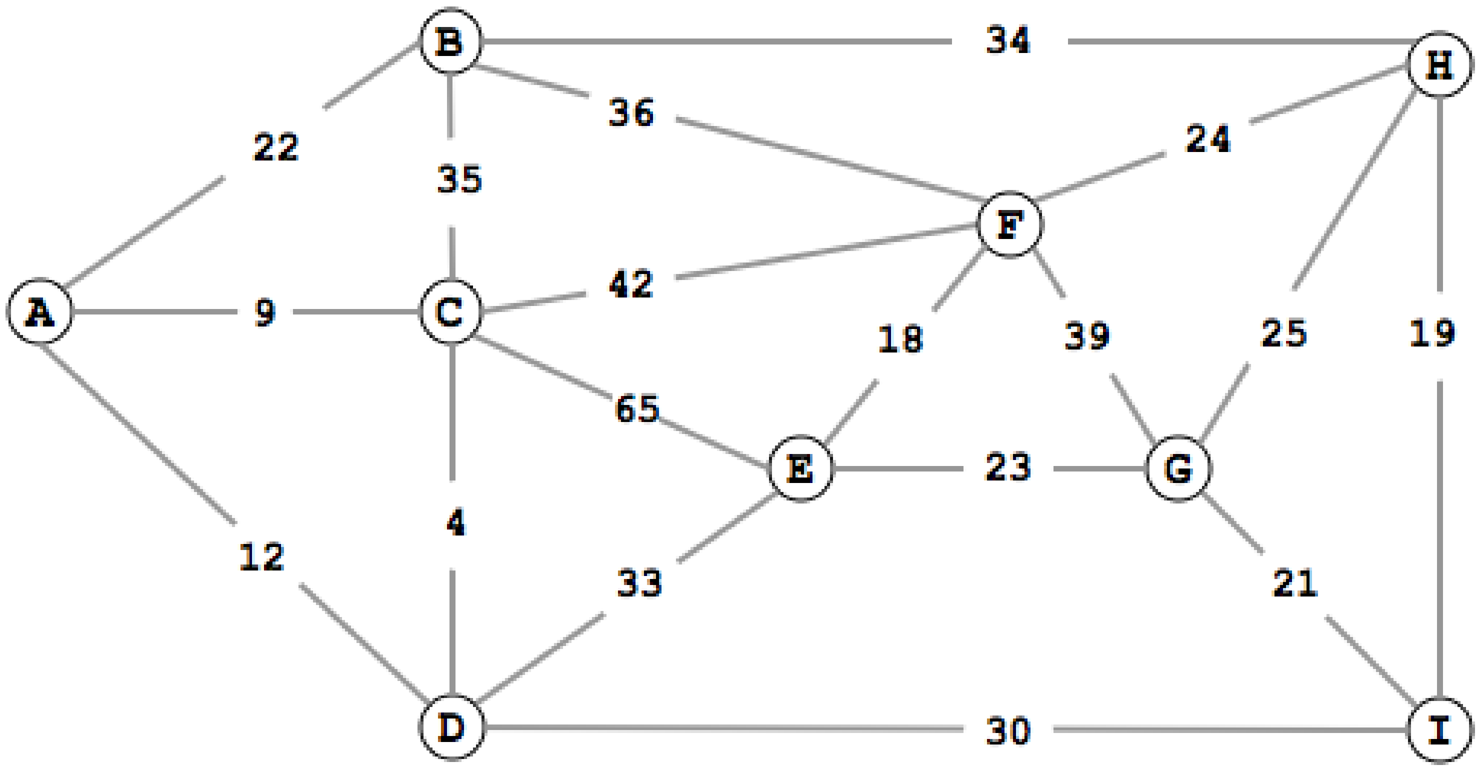
Cho $G = (V, E)$ là đồ thị có tập các đỉnh V và tập các cạnh E..

- ❖ Xuất phát từ 1 đỉnh tùy ý duyệt đỉnh này (gán nhãn v=1)và sau đó duyệt những đỉnh kề với đỉnh xuất phát.
- ❖ Duyệt xong một đỉnh thì đánh dấu đỉnh đã được duyệt.
- ❖ Ở bước tiếp theo ta duyệt tiếp tục duyệt đỉnh kề chưa được duyệt ở các đỉnh đã duyệt thứ 1,2,3… Cho đến khi mọi đỉnh đã được duyệt.

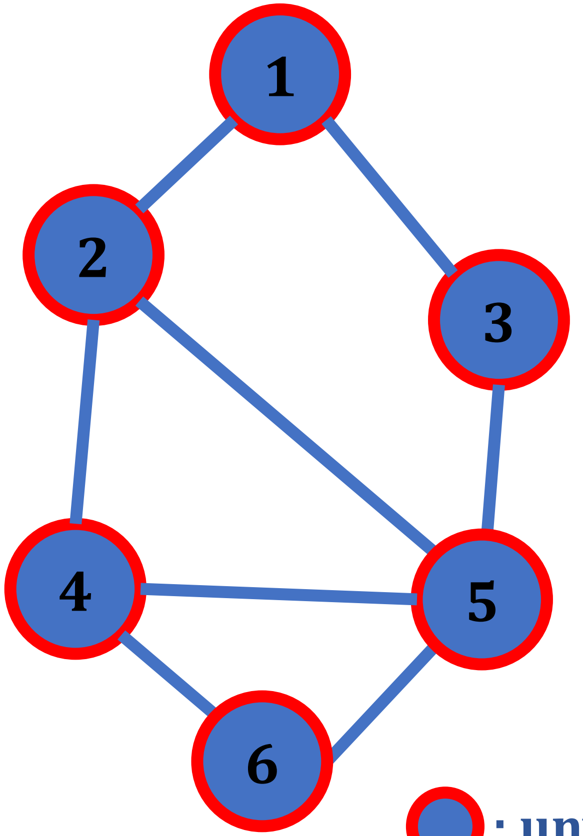
THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG (BFS)



THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU RỘNG (BFS)



MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)

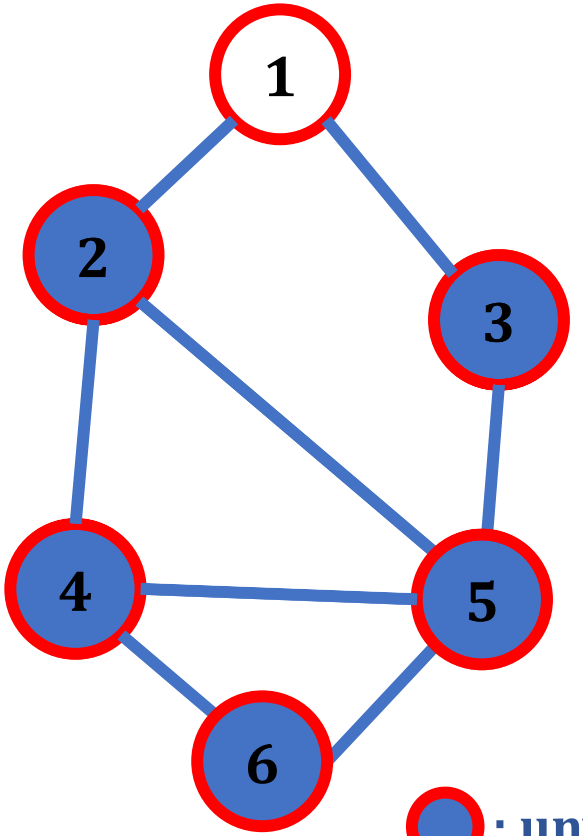


	1	2	3	4	5	6
QUEUE:	0	0	0	0	0	0

OUTPUT:

: unvisited   : visited

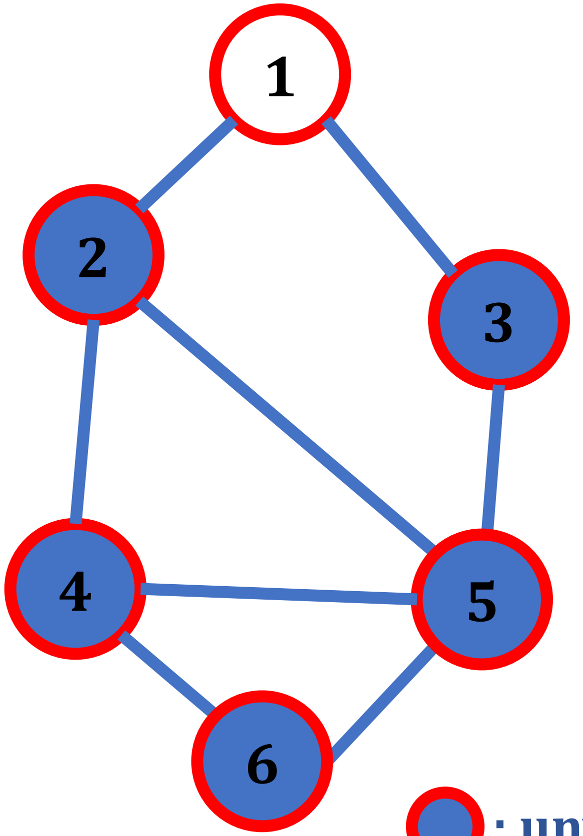
MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



	1	2	3	4	5	6
QUEUE:	1	0	0	0	0	0

OUTPUT: 1

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



	1	2	3	4	5	6
QUEUE:	1	0	0	0	0	0

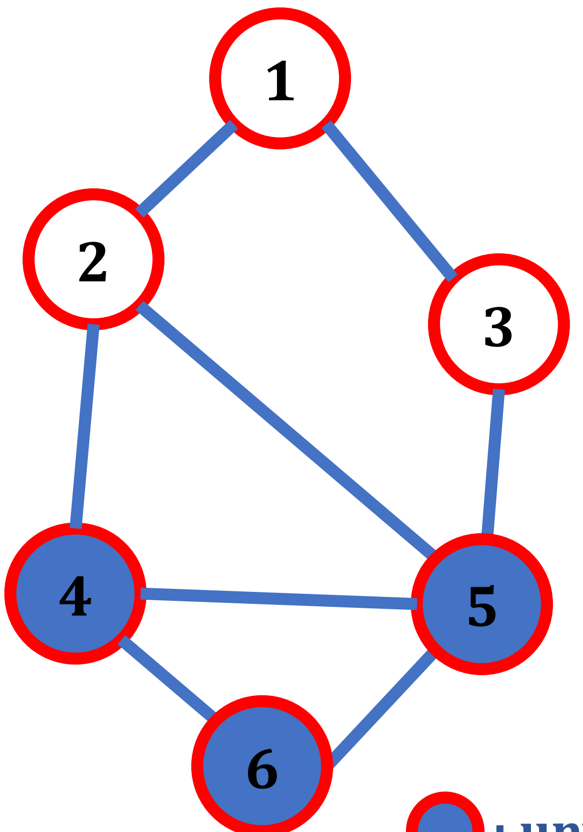
OUTPUT:

PRINT: 1

: unvisited

: visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



QUEUE:

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

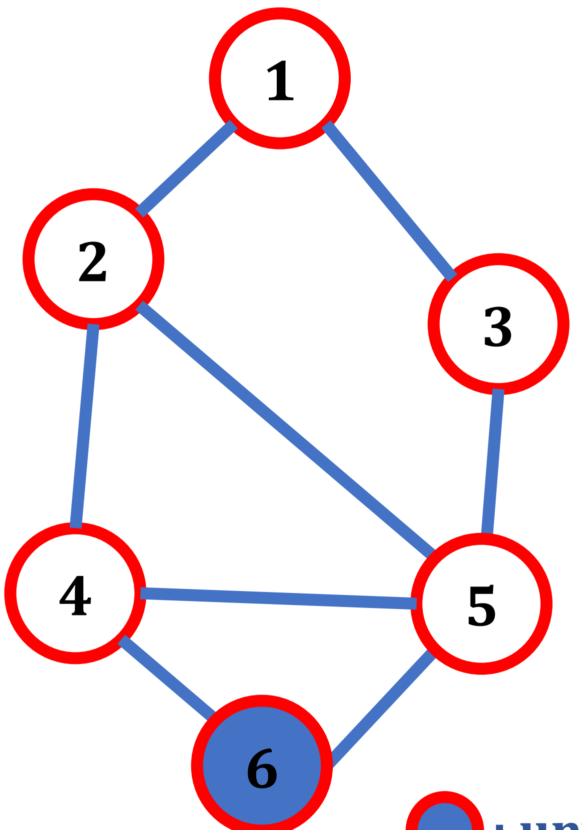
OUTPUT: 2 3

PRINT: 1

: unvisited

: visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



QUEUE:

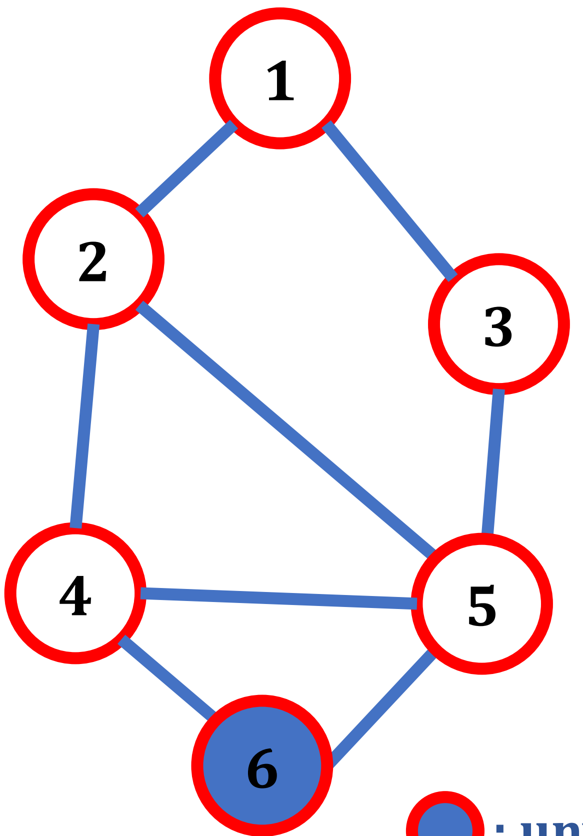
1	2	3	4	5	6
1	1	1	1	1	0

OUTPUT: 3 4 5

PRINT: 1 2

: unvisited　　: visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



QUEUE:

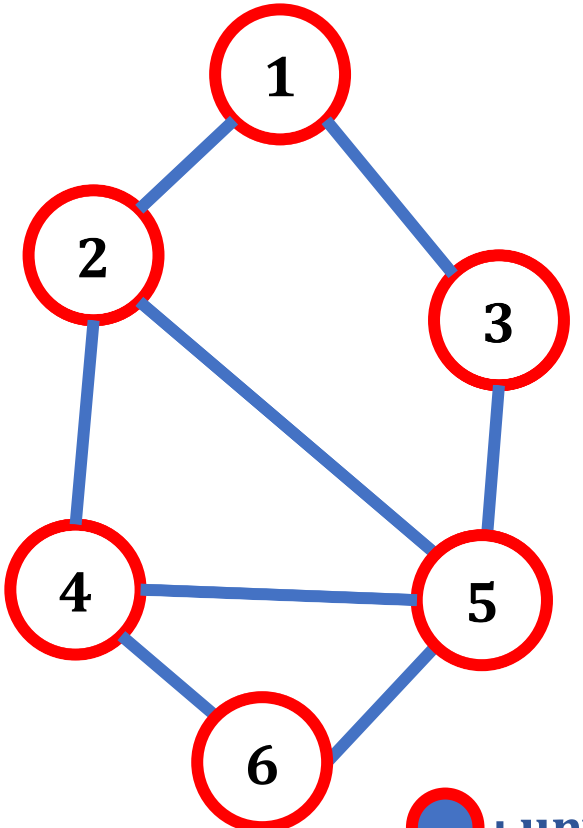
1	2	3	4	5	6
1	1	1	1	1	0

OUTPUT: 4 5

PRINT: 1 2 3

: unvisited : visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



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

QUEUE:

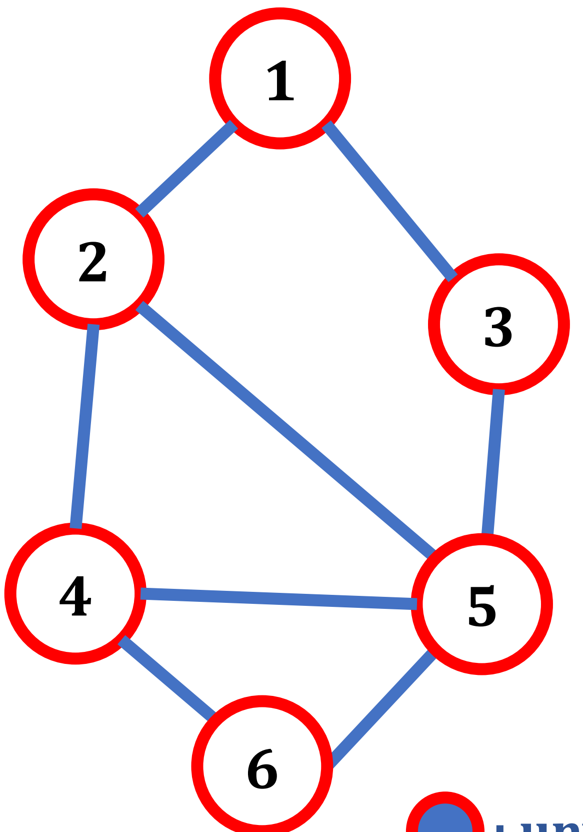
OUTPUT: 5 6

PRINT: 1 2 3 4

: unvisited   : visited

CÁC THUẬT TOÁN DUYỆT TRÊN ĐỒ THỊ

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



QUEUE:

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

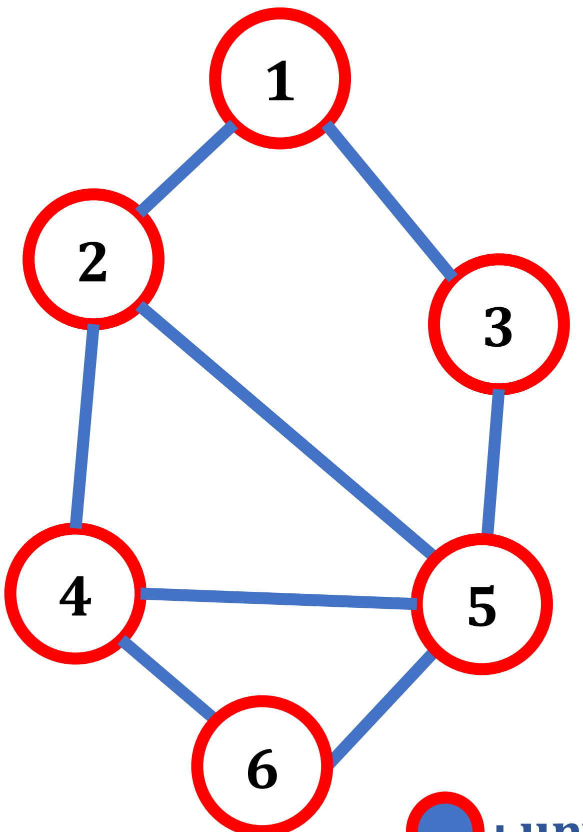
OUTPUT: 6

PRINT: 1 2 3 4 5

: unvisited

: visited

MINH HỌA THUẬT TOÁN DUYỆT ĐỒ THỊ THEO CHIỀU SÂU (BFS)



QUEUE:

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

OUTPUT:

PRINT: 1 2 3 4 5 6

: unvisited

: visited

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

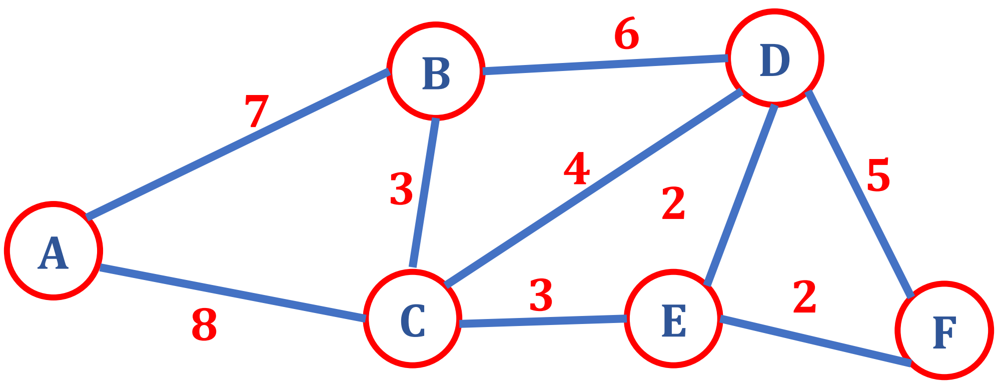
GIẢI THUẬT KRUSKAL:

- Kruskal là tìm cây khung nhỏ nhất dựa trên giải thuật tham lam. Giải thuật Kruskal xem đồ thị như là một rừng cây và mỗi nút là một cây riêng lẻ trong rừng.
- Kruskal không phụ thuộc vào điểm bắt đầu. Một cây kết nối với cây khác nếu và chỉ nếu cây này có **trọng số** nhỏ nhất trong số tất cả các cây đã tìm được.

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

- **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị

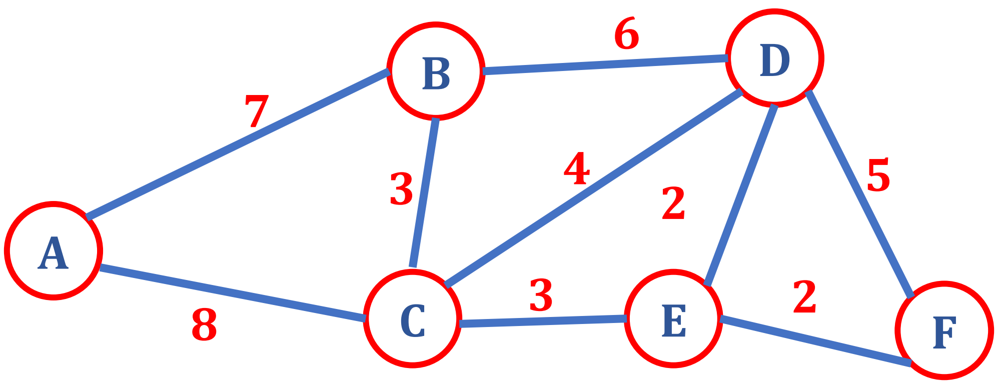


- **Bước 2:** Sắp xếp các cạnh theo trọng số tăng dần

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

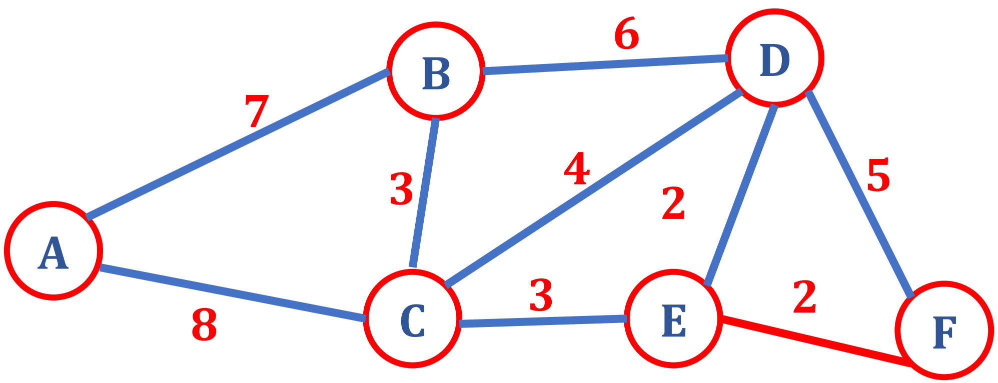


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất.

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

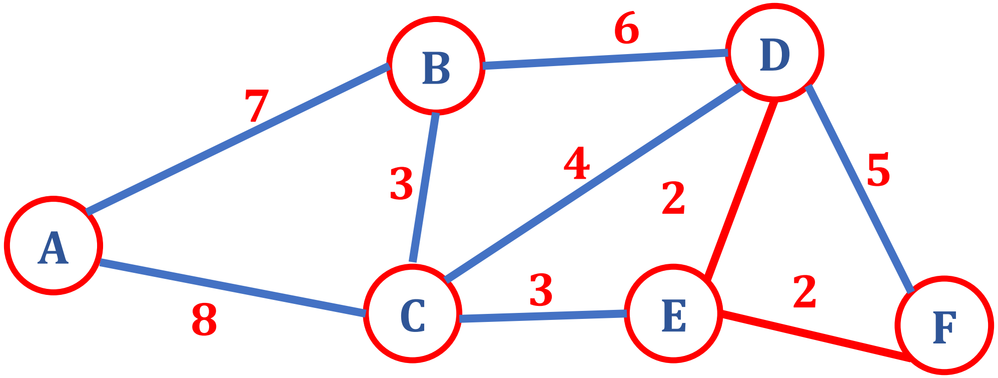


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất..... **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

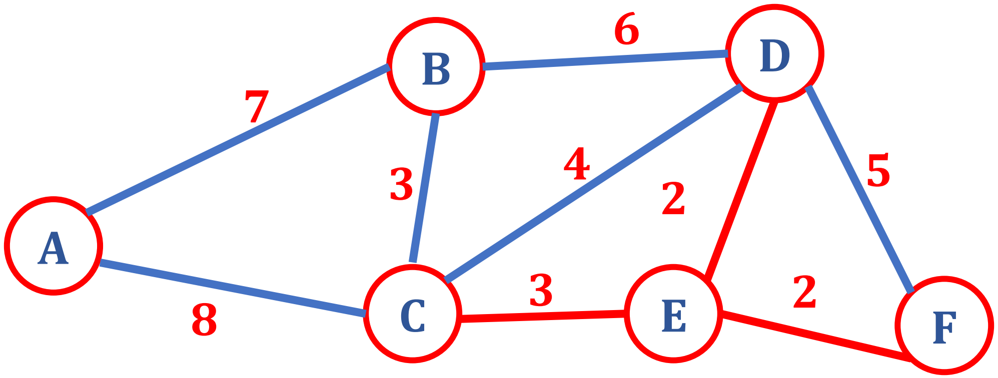


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất….. **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

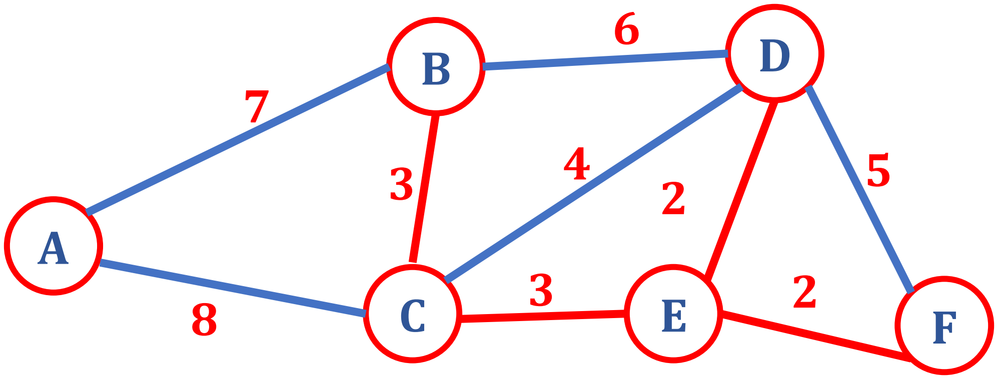


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất….. **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

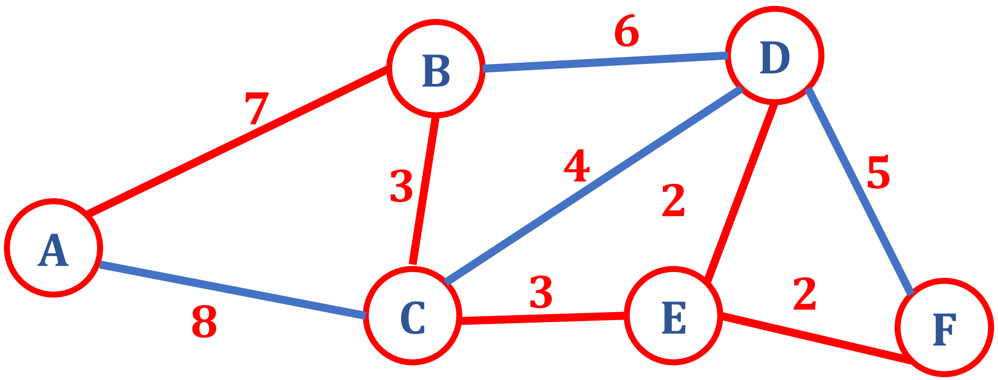


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất..... **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT KRUSKAL:

EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8

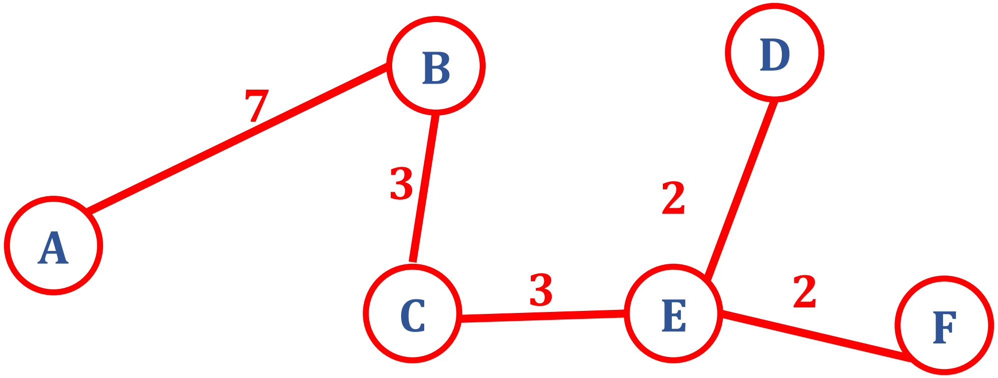


❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất..... **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

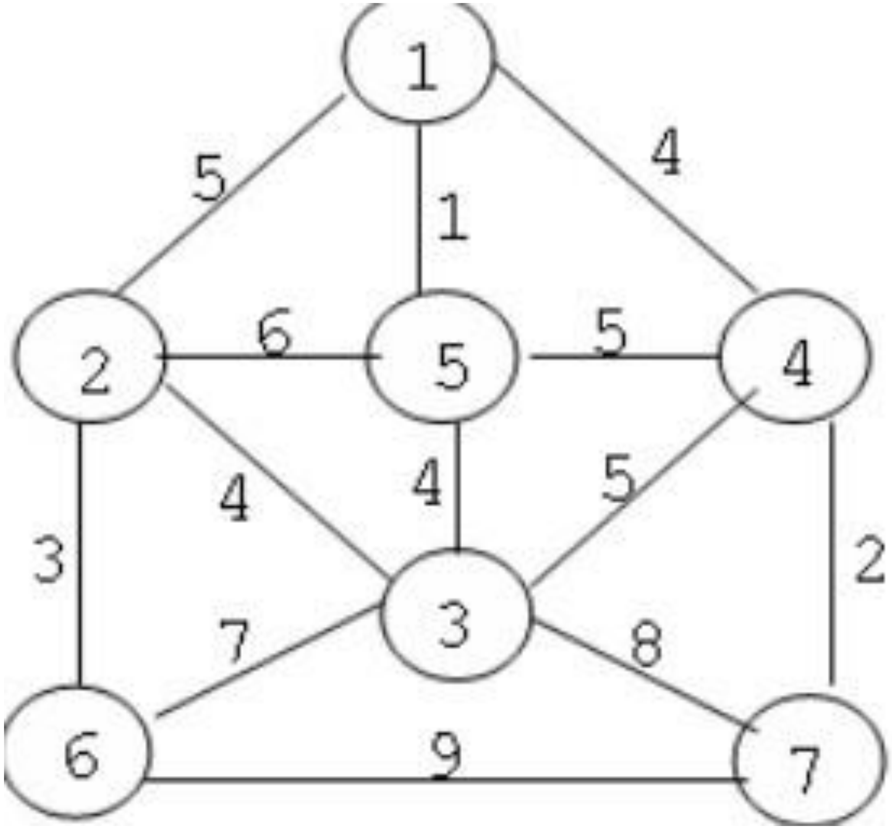
GIẢI THUẬT KRUSKAL:

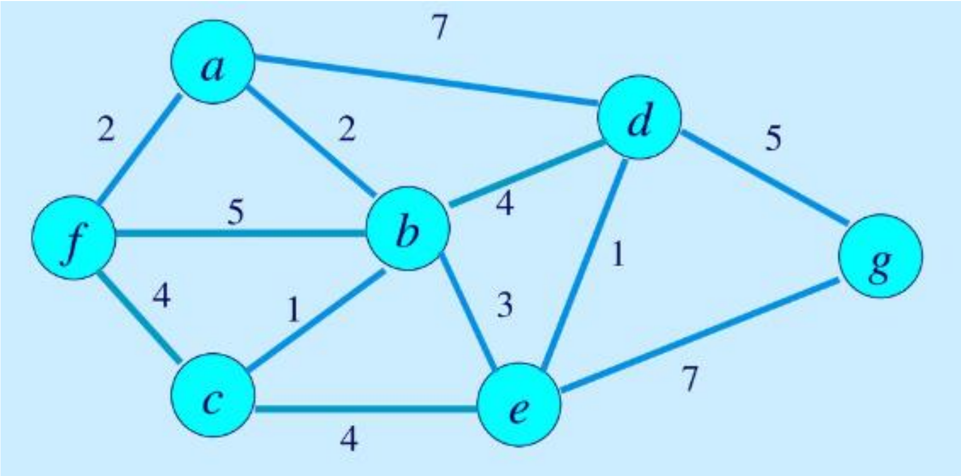
EF	DE	EC	BC	DC	DF	DB	AB	AC
2	2	3	3	4	5	6	7	8



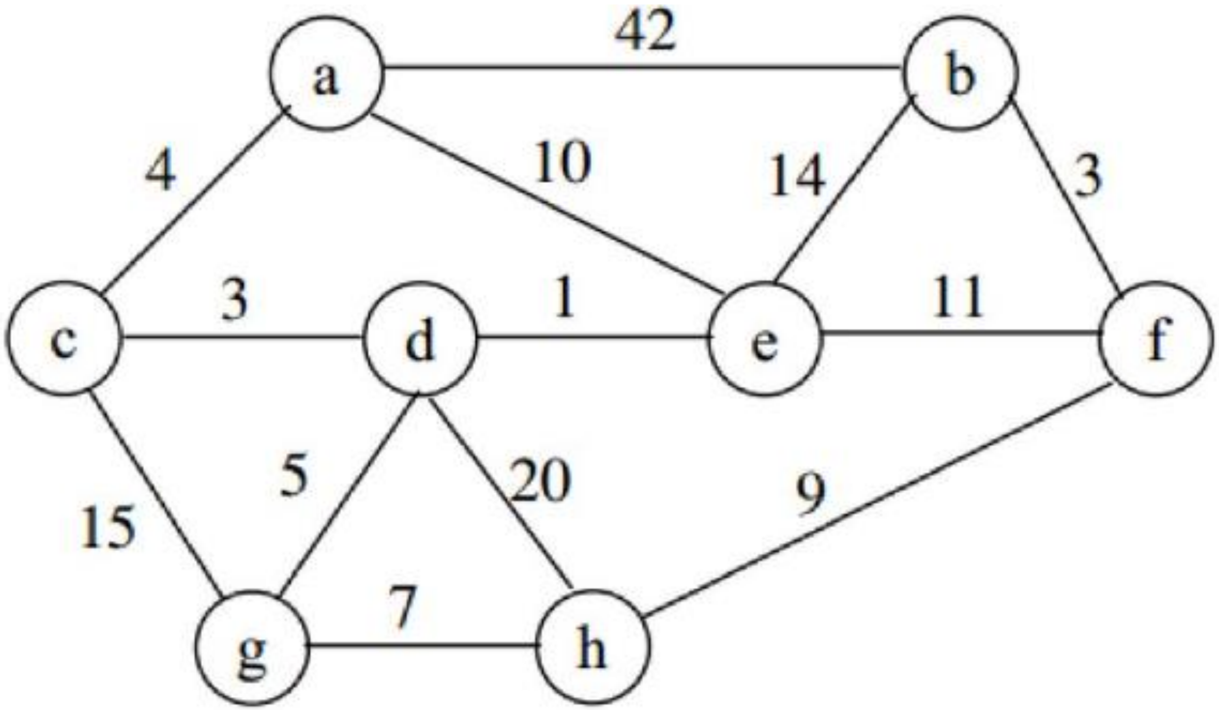
❖ **Bước 3:** Thêm 1 cạnh có trọng số nhỏ nhất….. **KIỂM TRA CÂY KHUNG KHÔNG TẠO CHU TRÌNH**

BÀI TẬP 1: TÌM CÂY KHUNG NHỎ NHẤT KRUSKAL





BÀI TẬP 2: TÌM CÂY KHUNG NHỎ NHẤT KRUSKAL



THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

- Giải thuật Prim, cũng giống như giải thuật Kruskal, là để tìm cây khung nhỏ nhất dựa vào giải thuật tham lam.
- Giải thuật Prim, trái ngược với giải thuật Kruskal, xem các nút như là một cây riêng là và vẫn cữ tiếp tục việc thêm các nút mới vào cây khung từ đồ thị đã cho.
- Giải thuật Prim phụ thuộc vào điểm bắt đầu.
- Minh họa giải thuật Prim chúng ta sẽ sử dụng cùng một ví du như trong giải thuật Kruskal.

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

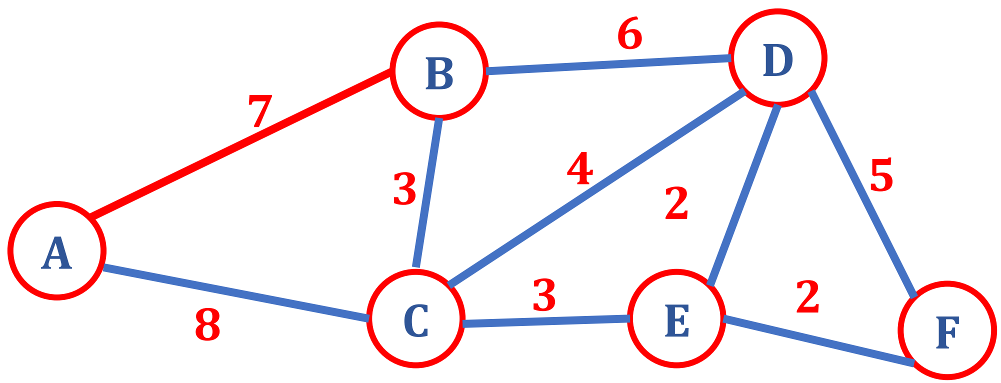
GIẢI THUẬT PRIM: Cho G=(X,E) là một đồ thị liên thông có trọng số gồm n đỉnh. Thuật toán Prim được dùng để tìm ra cây khung nhỏ nhất của G.

- **Bước 1**: Chọn tùy ý v thuộc V và khởi tạo Y:= {v}; T := Ø. Trong đó X là tập các đỉnh của đồ thị, Y là tập các đỉnh được chọn vào cây khung nhỏ nhất và T là tập các cạnh của cây này.
- **Bước 2**: Trong số những cạnh e nối đỉnh w với đỉnh v trong Y với w thuộc X\Y và v thuộc Y ta chọn cạnh có trọng lượng nhỏ nhất.
- **Bước 3**: Gán Y := Y thuộc {w} và T:= T thuộc {e}
- **Bước 4**: Nếu T đủ n – 1 phần tử thì dừng, ngược lại làm tiếp tục bước 2.

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

- **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị.

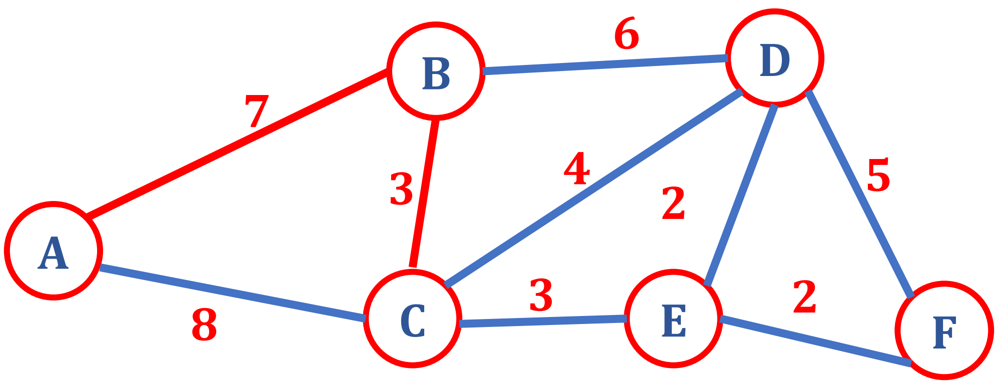


- **Bước 2:** Chọn node bất kỳ làm node gốc. Ví dụ chọn **A**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

- **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị.

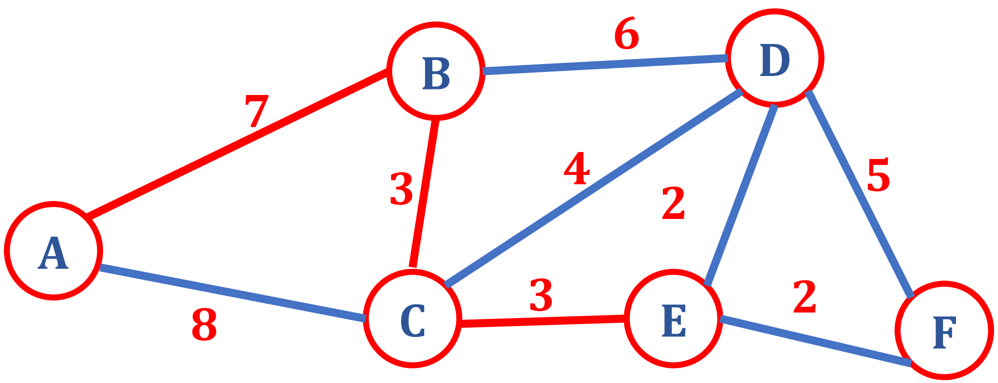


- **Bước 2:** Chọn node bất kỳ làm node gốc. Ví dụ chọn **A**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

❖ **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị.

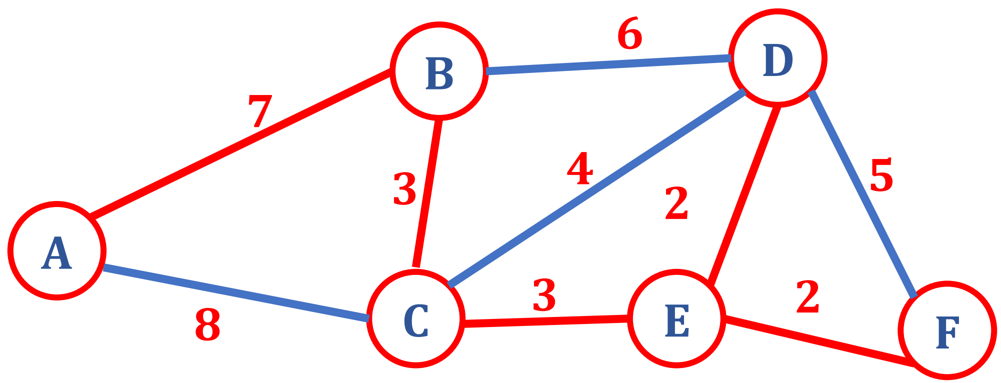


❖ **Bước 2:** Chọn node bất kỳ làm node gốc. Ví dụ chọn **A**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

- **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị.

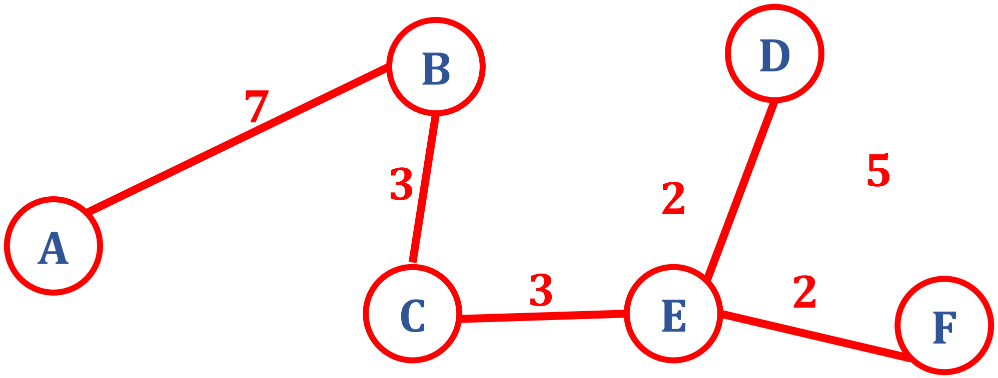


- **Bước 2:** Chọn node bất kỳ làm node gốc. Ví dụ chọn **A**

THUẬT TOÁN TÌM CÂY KHUNG NHỎ NHẤT

GIẢI THUẬT PRIM:

- **Bước 1:** Xóa tất cả các vòng và các cạnh song song, giữ các cạnh có trọng số nhỏ nhất trong đồ thị.

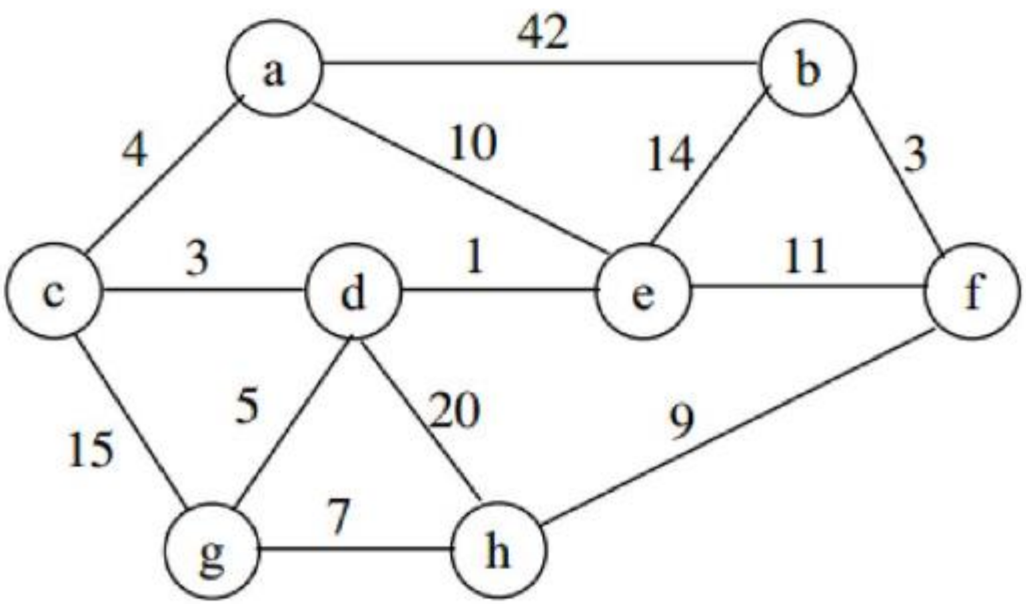


CÂY KHUNG NHỎ NHẤT

BÀI TẬP 1: TÌM CÂY KHUNG NHỎ NHẤT GIẢI THUẬT PRIM:

$$
\begin{array}{c}
\\ A \\ B \\ C \\ D \\ E \\ F \\ H \\ I
\end{array}
\begin{array}{c}
\begin{array}{cccccccc} A & B & C & D & E & F & H & I \end{array} \\
\begin{pmatrix}
\infty & 15 & 16 & 19 & 23 & 20 & 32 & 18 \\
15 & \infty & 33 & 13 & 34 & 19 & 20 & 12 \\
16 & 33 & \infty & 13 & 29 & 21 & 20 & 19 \\
19 & 13 & 13 & \infty & 22 & 30 & 21 & 11 \\
23 & 34 & 29 & 22 & \infty & 34 & 23 & 21 \\
20 & 19 & 21 & 30 & 34 & \infty & 17 & 18 \\
32 & 20 & 20 & 21 & 23 & 17 & \infty & 14 \\
18 & 12 & 19 & 11 & 21 & 18 & 14 & \infty
\end{pmatrix}
\end{array}.
$$

BÀI TẬP 2: TÌM CÂY KHUNG NHỎ NHẤT GIẢI THUẬT PRIM:



BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT (DIJSKTRA)

Dijkstra là thuật toán tìm đường đi ngắn nhất giữa hai đỉnh của một đồ thị có trọng số không âm.

Cho đồ thị đơn **G = (V, E)** có trọng số không âm gồm **N** đỉnh. Đường đi ngắn nhất từ **s** đến **g** sẽ được xác định như sau

$$d(u) = min\{d(v) + w(v,u), v \in X^{-}(u)\}$$