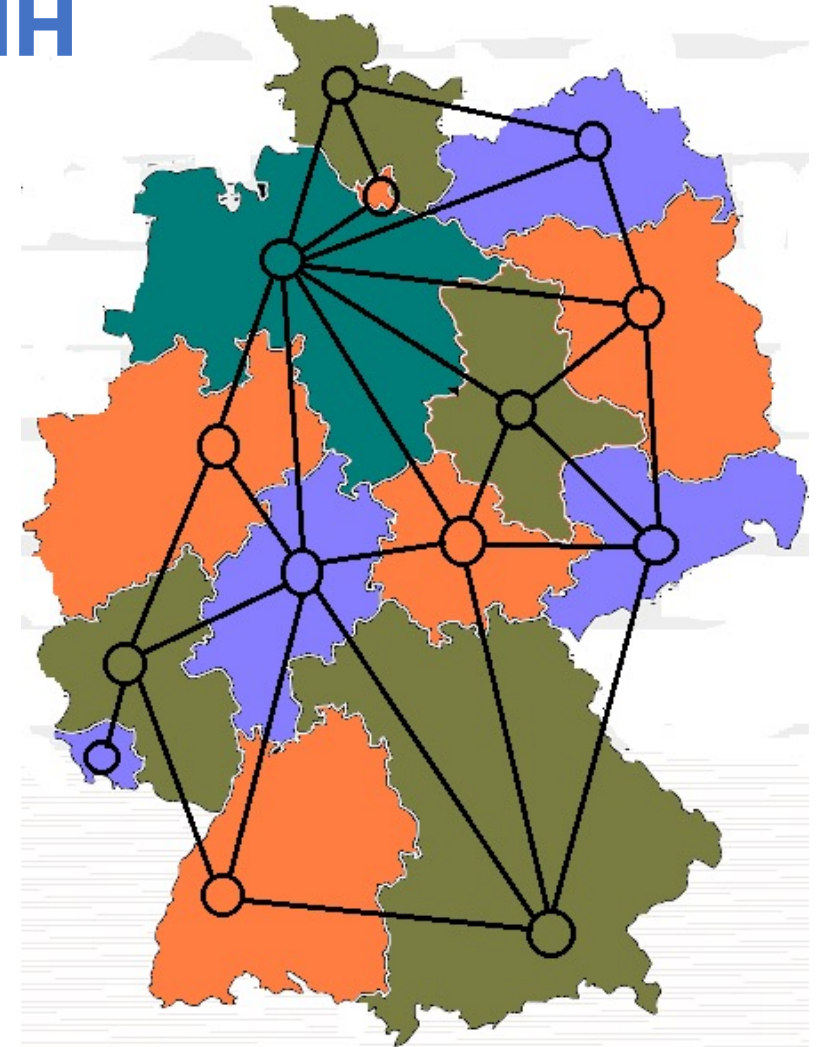


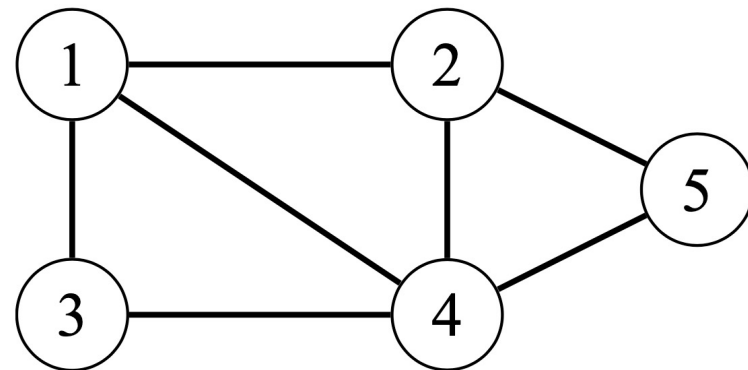
BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

Vũ Ngọc Thanh Sang
Khoa Công nghệ Thông tin
Đại học Sài Gòn



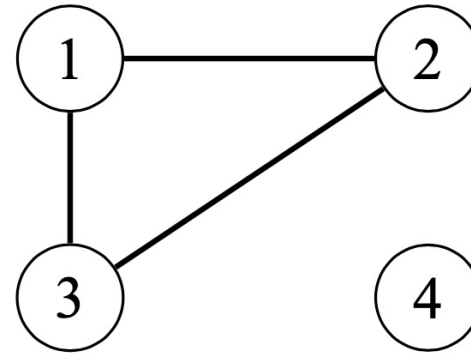
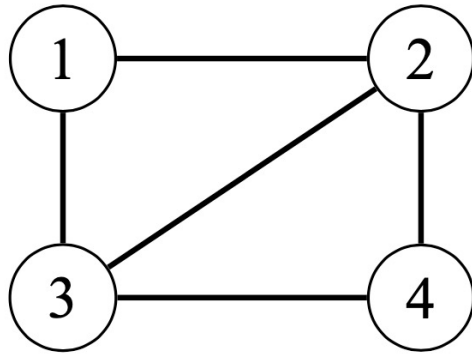
ÔN TẬP

- **Đồ thị** là cấu trúc toán học đại diện cho các mối quan hệ của các cặp đối tượng. Một đồ thị, kí hiệu là $G(V, E)$ gồm hai thành phần:
 - Tập hợp các **đỉnh** (vertex, V) đại diện cho các đối tượng
 - Tập hợp các **cạnh** (edge, E) bao gồm một cặp các đỉnh
- Hai điểm gọi là **lân cận** (neighbor, adjacent) nếu chúng được nối với nhau bởi một cạnh.
- **Đường đi** (path) là tập hợp các cạnh lần lượt đi qua các đỉnh của đồ thị.
- **Độ dài** của đường đi là số đỉnh nằm trên đường đi.

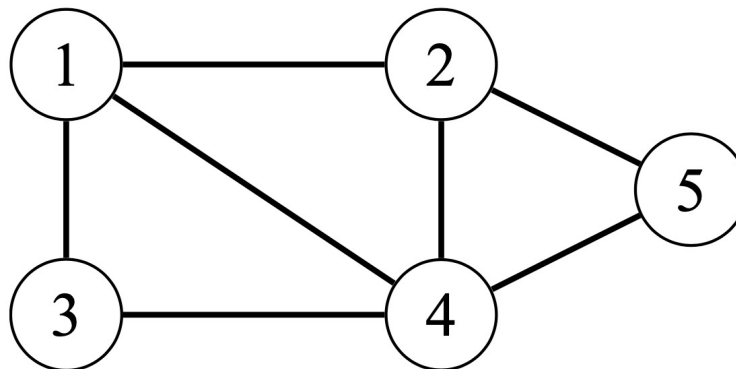


ÔN TẬP

- Đồ thị có liên kết nếu có cạnh ở giữa 2 nút (node).

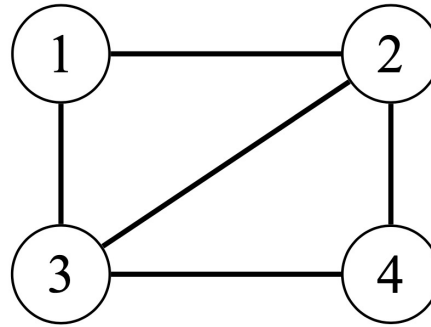


- Số **bậc** (degree) của đồ thị là số đỉnh có liên kết của đồ thị đó.

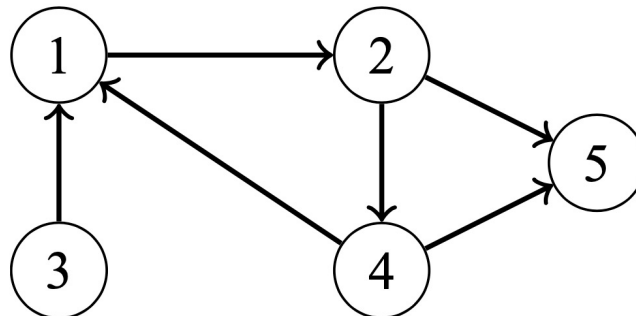


ÔN TẬP

1. Đồ thị vô hướng (undirected graph): là đồ thị trong đó tất cả các cạnh đều có hai hướng.

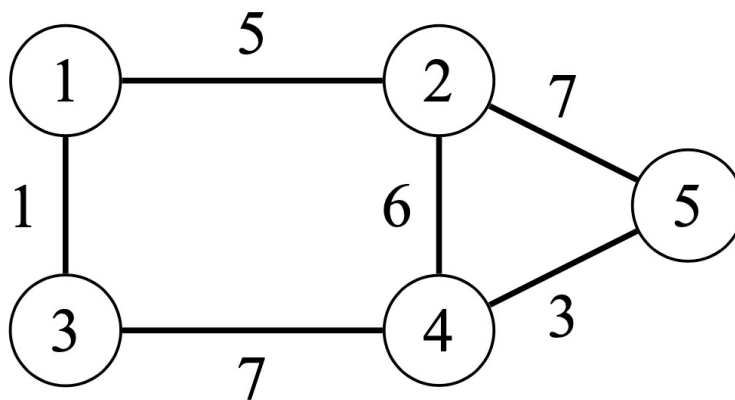


2. Đồ thị có hướng (directed graph): là đồ thị trong đó tất cả các cạnh đều chỉ về một hướng nhất định.

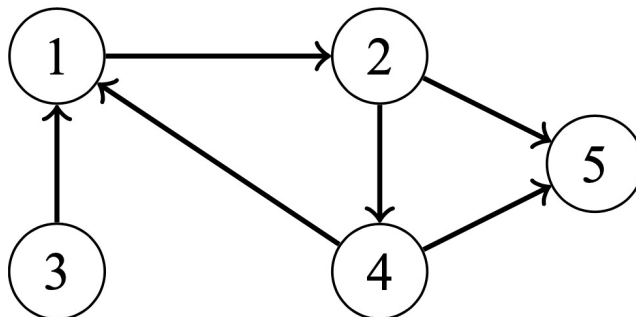


ÔN TẬP

3. Đồ thị có trọng số (weighted graph): mỗi cạnh của đồ thị được gán một trọng số, hoặc chi phí.



4. Đồ thị chu trình (cyclic graph): là đồ thị bao gồm một đường đi từ một đỉnh và kết thúc tại cùng một đỉnh. Đường đi đó gọi là một chu trình.



BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

- Việc lựa chọn cấu trúc dữ liệu phụ thuộc vào kích thước của đồ thị và cách mà các thuật toán sẽ xử lý dữ liệu.
 - Danh sách kề cận (Adjacency Lists)
 - Ma trận kề cận (Adjacency Matrix)
 - Danh sách đỉnh (Edge List)

BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

○ Danh sách kề cận (Adjacency Lists)

Để chứa các danh sách kề cận trong đồ thị, ta sử dụng vector với cú pháp

```
vector<int> adj[N];
```

Với N được chọn để chứa được tất cả các danh sách kề cận.

Ví dụ với $N = 4$

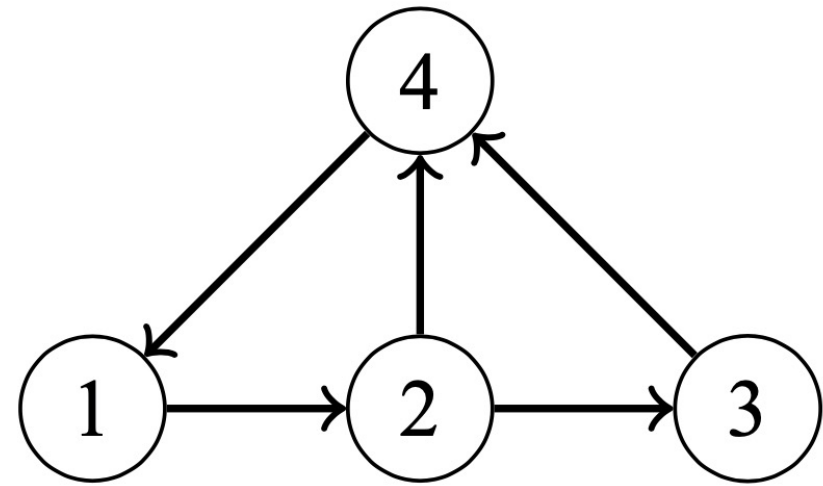
```
adj[1].push_back(2);
```

```
adj[2].push_back(3);
```

```
adj[2].push_back(4);
```

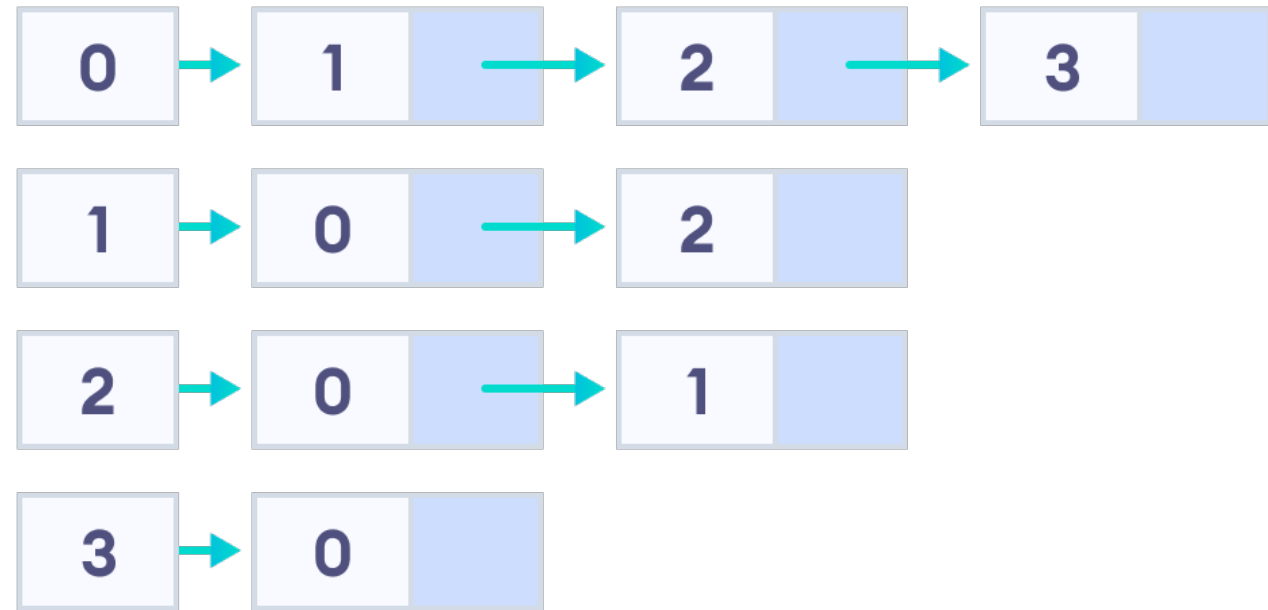
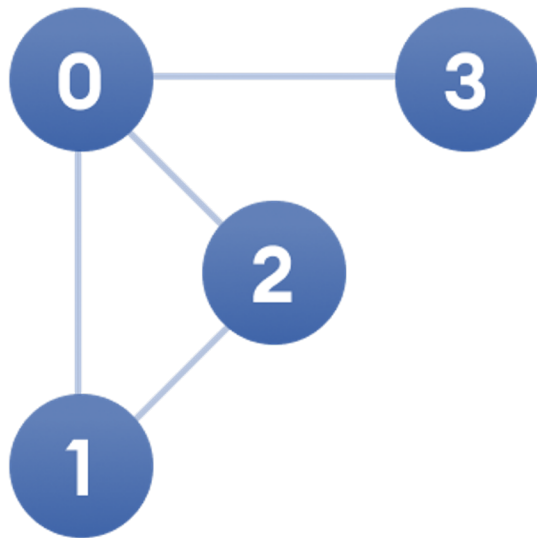
```
adj[3].push_back(4);
```

```
adj[4].push_back(1);
```



BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

- **Danh sách kề cận (Adjacency Lists)**



BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

○ Danh sách kề cận (Adjacency Lists)

Với đồ thị có trọng số, ta sử dụng vector với cú pháp

```
vector<pair<int,int>> adj[N];
```

Với N được chọn để chứa được tất cả các danh sách kề cận.

Ví dụ với $N = 4$

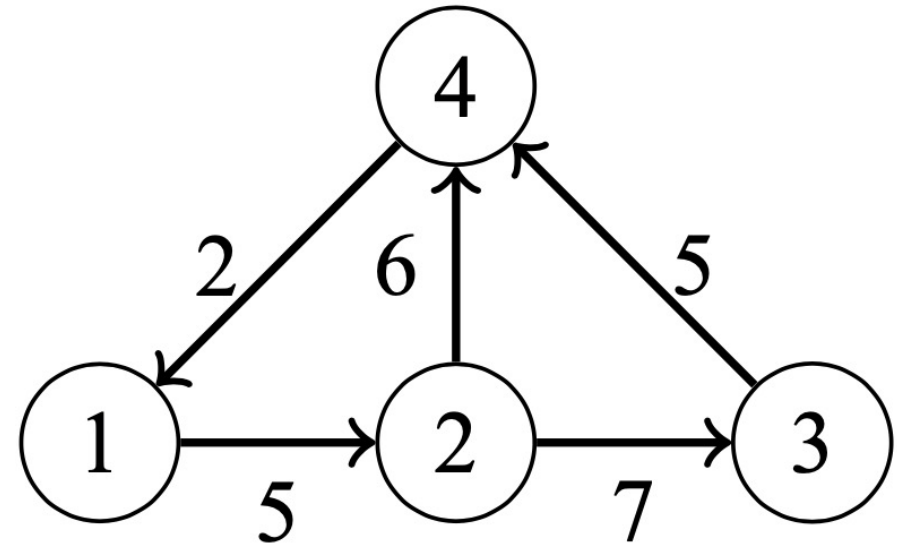
```
adj[1].push_back(2,5);
```

```
adj[2].push_back(3,7);
```

```
adj[2].push_back(4,6);
```

```
adj[3].push_back(4,5);
```

```
adj[4].push_back(1,2);
```



BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

- **Danh sách kề cận (Adjacency Lists)**

Để liệt kê và xử lý các nút kề cận nút s ta sử dụng cú pháp:

```
for (auto u : adj[s]) {  
    // process node u  
}
```

BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

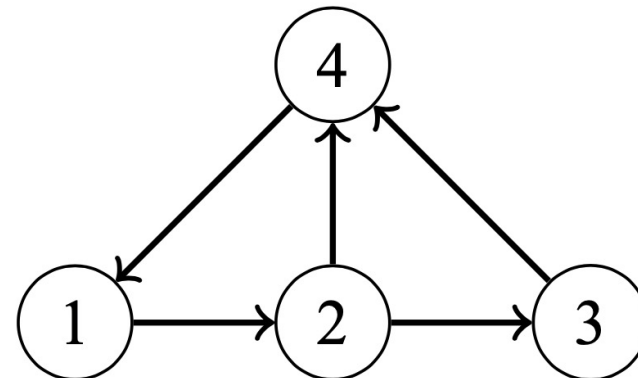
○ Ma trận kề cận (Adjacency Matrices)

- Ma trận kề cận cho biết các cạnh của đồ thị. Từ ma trận, chúng ta có thể kiểm tra một cách hiệu quả nếu tồn tại một cạnh giữa hai nút.

```
int adj[N][N];
```

- Trong đơn đồ thị, mỗi giá trị $\text{adj}[a][b]$ cho biết đồ thị có chứa cạnh từ nút a tới node b hay không. Nếu có thì $\text{adj}[a][b] = 1$, nếu không thì $\text{adj}[a][b] = 0$.
- Trong đa đồ thị, $\text{adj}[a][b] = k$ với k là số cạnh liên kết 2 đỉnh a, b

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

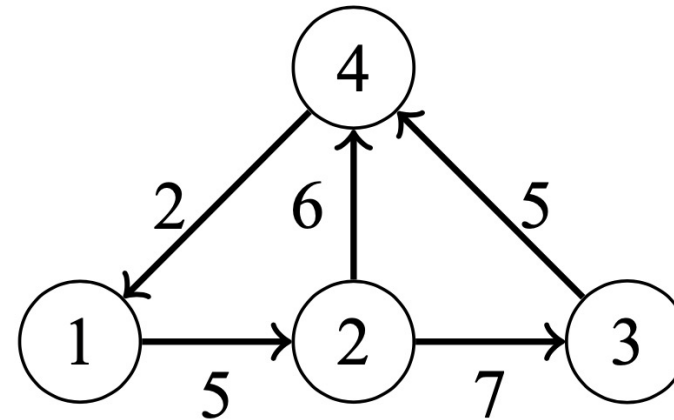


BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

- **Ma trận kề cận (Adjacency Matrices)**

Nếu ma trận có trọng số, chúng ta sử dụng giá trị $\text{adj}[a][b]$ để diễn tả trọng số của cạnh (nếu tồn tại) từ node a tới node b.

$$\begin{bmatrix} 0 & 5 & 0 & 0 \\ 0 & 0 & 7 & 6 \\ 0 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$



Nhược điểm của ma trận kề cận là bao gồm n^2 phần tử và đa số là 0. Vì vậy không nên sử dụng ma trận kề cận cho đồ thị lớn.

BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

○ Danh sách đỉnh (Edge List)

Một danh sách chứa tất cả các cạnh của đồ thị theo một thứ tự nhất định. Đây là cách tiện lợi để biểu diễn đồ thị nếu thuật toán xử lý tất cả các cạnh mà không cần phải tìm cạnh bắt đầu tại một node nhất định.

```
vector<pair<int,int>> adj[N];
```

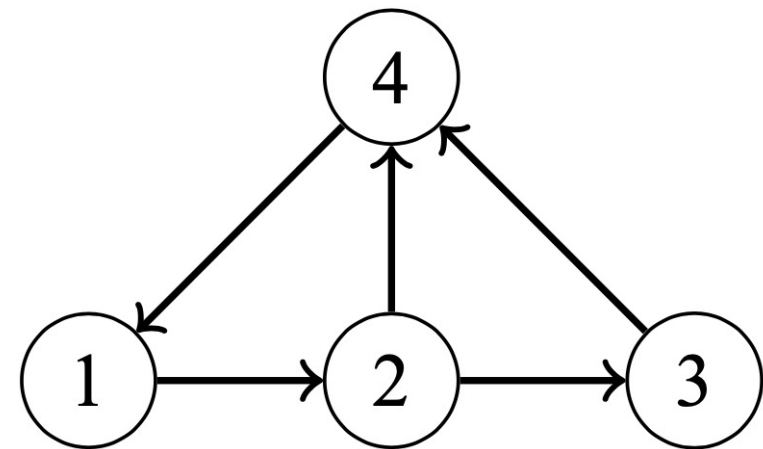
```
edges.push_back({1,2});
```

```
edges.push_back({2,3});
```

```
edges.push_back({2,4});
```

```
edges.push_back({3,4});
```

```
edges.push_back({4,1});
```



BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH

- **Danh sách đỉnh (Edge List)**

Nếu đồ thị có trọng số, ta sử dụng cú pháp sau

```
vector<tuple<int,int,int>> adj[N];
```

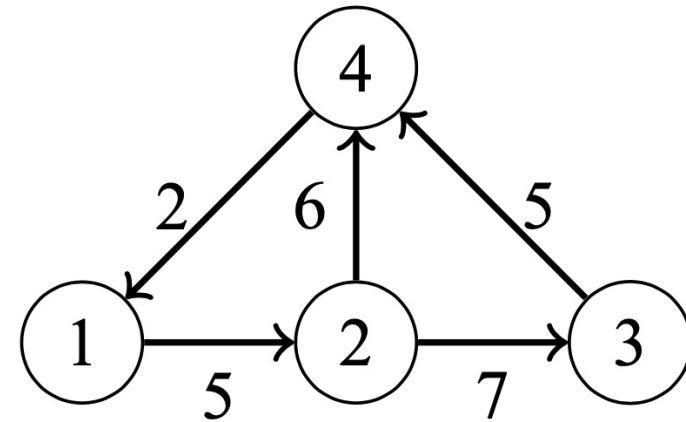
```
edges.push_back({1,2,5});
```

```
edges.push_back({2,3,7});
```

```
edges.push_back({2,4,6});
```

```
edges.push_back({3,4,5});
```

```
edges.push_back({4,1,2});
```



THANK YOU FOR YOUR ATTENTION.