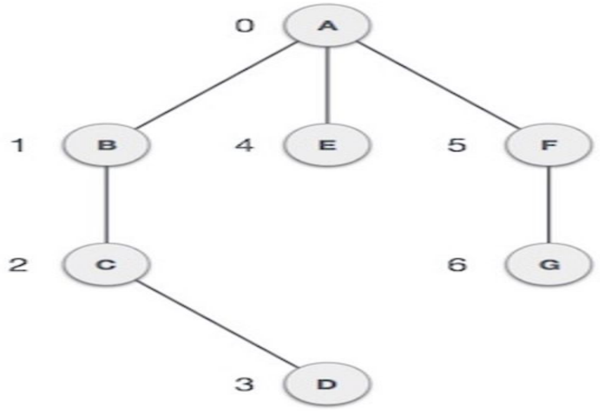
# Lab 3
# Implementing Minimum spanning tree(Prim's Algorithm)
# &
# Shortest Path Algorithm(Dijkstra's Algorithm)

The goal of this lab is to become familiar with the graph data structure and its application.

**Graph**: is a non-empty finite set V of elements called vertices together with a possibly empty set E of pairs of vertices called edges. G(V,E)



# 1. Minimum Spanning Tree

## 1.1 Introduction

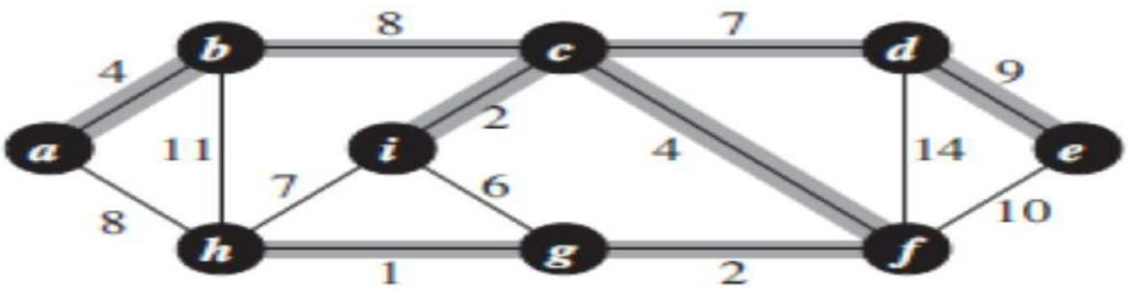
Theorem: Every connected graph has a spanning tree.

**Weighted Graphs**: A weighted graph is a graph, in which each edge has a weight (some real number).
**Weight of a Graph**: The sum of the weights of all edges.

A Minimum Spanning Tree in an undirected connected weighted graph is a spanning tree of minimum weight (among all spanning trees).

The minimum spanning tree may not be unique. However, if the weights of all the edges are pairwise distinct, it is indeed unique.

MST Problem: Given a connected weighted undirected graph G , design an algorithm that outputs a minimum spanning tree (MST) of G.



## Prim’s Algorithm

The Prim’s algorithm makes a nature choice of the cut in each iteration – it grows a single tree and adds a light edge in each iteration.

Check visualization for more understanding:
https://www.cs.usfca.edu/~galles/visualization/Prim.html

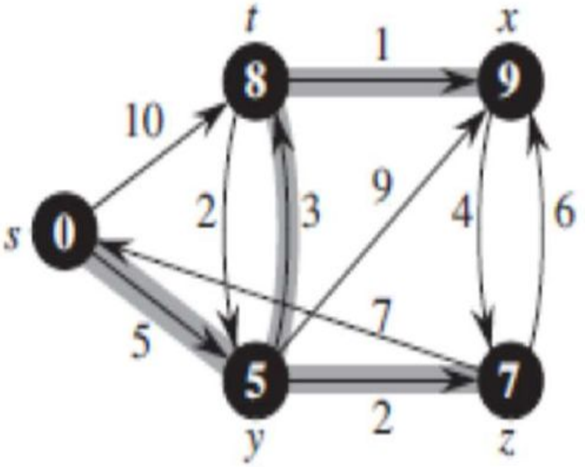
### 1.2 Requirements

Given: Weighted undirected graph, single source s.
Goal: Find Edges (MST) using Prim’s algorithm starting from s to include all vertex

You're required to implement the above procedures, pseudo code for the above procedures are explained in details in tutorials

## 2. Shortest Path Algorithm

The shortest path between two vertices is a path with the shortest length(weight).



### Dijkstra’s Algorithm

The Dijkstra’s algorithm is a solution to the single-source shortest path problem in graph theory !
Both directed and undirected graphs !
All edges must have nonnegative weights !
Graph must be connected

Check visualization for more understanding:
https://www.cs.usfca.edu/~galles/visualization/Dijkstra.html

### 2.2 Requirements

Given: Weighted directed graph with **positive** edge weights, single source s.

Goal: Determine the distance and a shortest path from the source vertex to every other vertex in the directed graph using Dijkstra's algorithm.

You're required to implement the above procedures, pseudo code for the above procedures are explained in details in tutorials

## 3. Notes

- Implement your algorithms using (Java or C/C++)
- You should work in groups **of 2 members**
- Discussion will have higher weight than implementation, so you should understand your implementation well to get discussion marks.

**Good Luck**