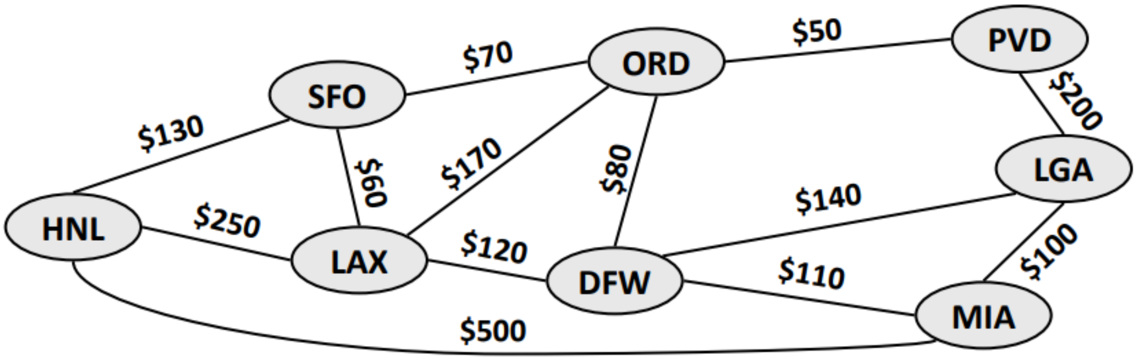
CSE- 3004 LAB-5 Assignment

Academic year: 2020-2021 **Semester:** WIN

Faculty Name: Dr. D Sumathi Mam **Date:** 10 /27/2021

Student name: M.Taran **Reg. no.:** 19BCE7346

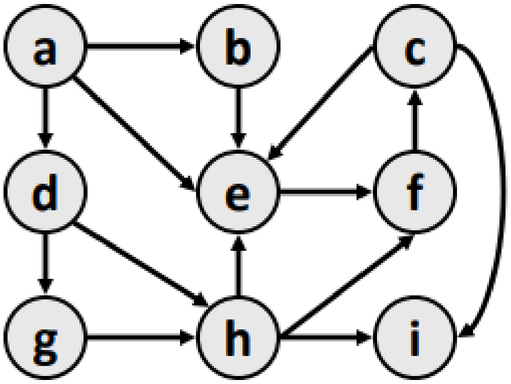
1. Find the minimum cost of travelling from SFO to your choice of place. Write a Java program to implement it.



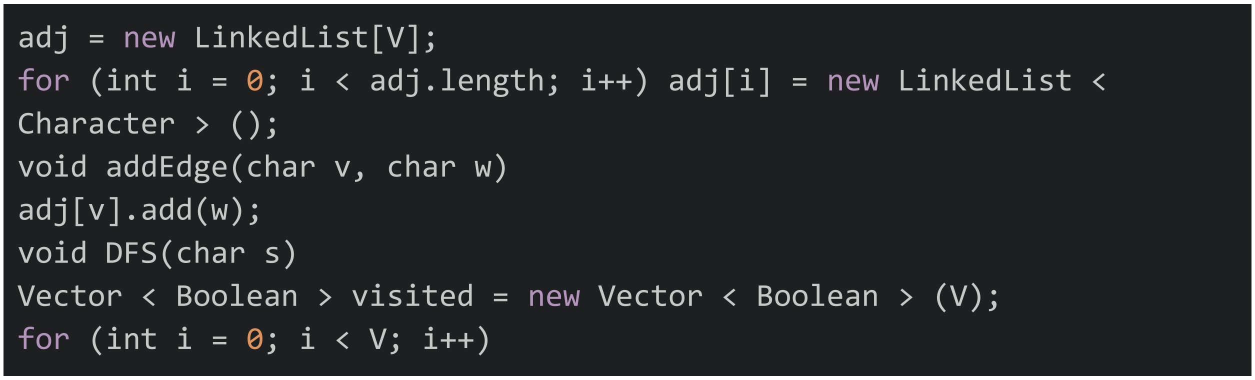
```
import java.util.*;
import java.lang.*;
public class Main {
        static int V = 15;
        static int INF = Integer.MAXVALUE;
        static int minimumCostSimplePath(int u, int destination,
boolean visited[], int graph[][])
        if (u destination) return 0;
        visited[u] = true;
        int ans = INF;
        for (int i = 0; i < V; i++)
            if (graph[u][i] != INF && !visited[i])
                int curr = minimumCostSimplePath(i, destination,
visited, graph);
        if (curr < INF)
            ans=Math.min(ans, graph[u][i] + curr);
        visited[u] = false;
        return ans;
        public static void main(String[] args)
        int graph[][] = new int[V][V];
        for (int i = O; i < V; i++)
            for (intj = 0; j < V; j++)
                graph[i][j] = INF;
        boolean visited[] = new boolean[V]; = 130
           graph[0][1] = 130;
           graph[0][2] = 60;
           graph[0][3] = 70;
           graph[0][0] = 0;
           graph[1][0] = 130;
           graph[1][1] = 0;
           graph[1][2] = 250;
           graph[1][7] = 500;
           graph[2][0] = 60;
           graph[2][1] = 250;
           graph[2][2] = O;
           graph[2][3] = 170;
           graph[2][4] = 120;
           graph[3][0] = 170;
           graph[3][3] = 0;
           graph[3][2] = 170;
           graph[3][4] = 80;
           graph[3][5] = 50;
           graph[4][2] = 120;
           graph[4][3] = 80;
           graph[4][4] = 140;
           graph[4][7] = 50;
           graph[4][6] = 140;
           graph[5][3] = 50;
           graph[5][6] = 200;
           graph[5][5] = 0;
           graph[6][4] = 140;
           graph[6][5] = 200;
           graph[6][7] = 100;
           graph[6][6] = 0;
           graph[7][4] = 110;
           graph[7][6] = 100;
           graph[7][7] = 0;

       int s = 0, t = 2;
    visited[s] = true;
    System.out.println(minimumCostSimplePath(s, t,visited, graph));
```

2. Implement DFS for the graph in Java



```
import java.util.*;
public class Main
static class Graph
int V;
LinkedList < Character > [] adj;
public Graph(int V)
this.V = V;
```



```
adj = new LinkedList[V];
for (int i = 0; i < adj.length; i++) adj[i] = new LinkedList <
Character > ();
void addEdge(char v, char w)
adj[v].add(w);
void DFS(char s)
Vector < Boolean > visited = new Vector < Boolean > (V);
for (int i = 0; i < V; i++)
```