ASSIGNMENT 2

NAME: Mujeeb ur Rahman khan

Roll no: 20736

CODE

/\*

Write a program to implement Bellman-Ford Algorithm using Dynamic Programming and verify the time complexity.

\*/

#include<bits/stdc++.h>

using namespace std;

int bellmanFord(int n, int m, int src, int dest, vector<vector<int>> edge);

int main() {

    int n;

    int src,dest;

    cout<<"enter number of nodes ";

        cin>>n;

    int m = n-1;

    cout<<endl<<"enter src ";

        cin>>src;

    cout<<endl<<"enter destination ";

        cin>>dest;

    vector<vector<int>> edge;

    for(int i=0;i<n;i++) {

        cout<<endl<<"Enter start node, end node, weight ";

        int p,q,r;

        cin>>p>>q>>r;

        edge.push\_back({p,q,r});

    }

    // edge.push\_back({1,2,2});

    // edge.push\_back({2,3,-1});

    // edge.push\_back({3,1,2});

    int ans = bellmanFord(n,m,src,dest,edge);

    cout<<endl<<"sortest distance between "<<src<<" and "<<dest<<" is "<<ans;

    return 0;

}

int bellmanFord(int n, int m, int src, int dest, vector<vector<int>> edges) {

    vector<int> dis(n+1,1e9);

    dis[src] = 0;

    for(int i=1;i<=m;i++){

        for(int j=0;j<n;j++){

                int u = edges[j][0];

                int v = edges[j][1];

                int w = edges[j][2];

                if(dis[u] != 1e9 && ((dis[u] + w) < dis[v]))

                    dis[v] = dis[u] + w;

            }

    }

    return dis[dest];

}

OUTPUT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OUTPUT\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

enter number of nodes 3

enter src 1

enter destination 3

Enter start node, end node, weight 1 2 2

Enter start node, end node, weight 2 3 -1

Enter start node, end node, weight 3 1 2

sortest distance between 1 and 3 is 1

Enter length of array