Djikstras Code

#include <bits/stdc++.h>

using namespace std;

int main()

{

cout << "enter the number of edges\n";

int n; cin >> n;

vector<vector<int>> v(n, vector<int>(3));

vector<vector<pair<int, int>>> adj(100);

cout << "Enter the edges with weight\n";

for(int i = 0; i < n; i++)

{

cin >> v[i][0] >> v[i][1] >> v[i][2];

adj[v[i][0]].push\_back({v[i][1], v[i][2]});

adj[v[i][1]].push\_back({v[i][0], v[i][2]});

}

// Adj list rep

vector<int> dist(100, INT\_MAX);

set<pair<int,int>> s;

int src;

cout << "Enter the source node\n";

cin >> src;

dist[src] = 0;

s.insert({0, src});

int c = 0;

while(!s.empty())

{ c++;

auto it = s.begin();

int distTillNow = (\*it).first;

int currNode = (\*it).second;

s.erase(it);

for(auto &i: adj[currNode])

{

int nbr = i.first;

int cost = i.second;

if(distTillNow + cost < dist[nbr])

{

auto f = s.find({dist[nbr], nbr});

if(f != s.end())

{

s.erase(f);

}

dist[nbr] = distTillNow + cost;

s.insert({distTillNow + cost, nbr});

}

}

cout << "AFTER ITERATION " << c << "\n";

for(int i = 0; i < 9; i++)

{

cout << i << ": " << dist[i] << "\n";

}

cout << "\n\n";

}

}

USED ADJ LIST;

