**Prim's MST**

#include <bits/stdc++.h>

using namespace std;

vector<pair<pair<int, int>, int>> calculatePrimsMST(int n, int m, vector<pair<pair<int, int>, int>>& edges) {

unordered\_map<int, list<pair<int, int>>> adj;

for (int i = 0; i < edges.size(); i++) {

int u = edges[i].first.first;

int v = edges[i].first.second;

int w = edges[i].second;

adj[u].push\_back(make\_pair(v, w));

adj[v].push\_back(make\_pair(u, w));

}

vector<int> key(n + 1);

vector<bool> visited(n + 1);

vector<int> parent(n + 1);

for (int i = 1; i <= n; i++) {

key[i] = INT\_MAX;

parent[i] = -1;

visited[i] = false;

}

key[1] = 0;

parent[1] = -1;

for (int i = 1; i < n; i++) {

int minKey = INT\_MAX;

int u;

for (int v = 1; v <= n; v++) {

if (visited[v] == false && key[v] < minKey) {

u = v;

minKey = key[v];

}

}

visited[u] = true;

for (auto it : adj[u]) {

int v = it.first;

int weight = it.second;

if (visited[v] == false && weight < key[v]) {

parent[v] = u;

key[v] = weight;

}

}

}

vector<pair<pair<int, int>, int>> result;

for (int i = 2; i <= n; i++) {

result.push\_back({{parent[i], i}, key[i]});

}

return result;

}