Second-Best-MST

```
#include<bits/stdc++.h>
#include<algorithm>
#include<iostream>
#include<limits.h>
#include<string.h>
#include <utility>
#include <cstdio>
#include <algorithm>
#include <vector>
using namespace std;

int t, cs=0, n, e, src, dest, cost, sz;
vector<int> pset(1100);
vector<pair<int, pair<int, int> > > edgeList;
vector<int> tree;

void initialize(){
    for(int i=0; i<n; i++)
        pset[i] = i;
    sz = n;
}

int findSet(int i){
    return pset[i] == i ? i : (pset[i] = findSet(pset[i]));
}

int isSameSet(int i, int j){
    return findSet(i) == findSet(j);
}

void unionSets(int i, int j){
    if (!isSameSet(i, j)){
        pset[findSet(i)] = findSet(j);
        sz--;
    }
}

int kruskal(int index) {
    int mst = 0;
    initialize();

    for(int i=0; i<edgeList.size() && sz>1; i++){
        pair<int, pair<int, int> > front = edgeList[i];
        int a, b;
        a = front.second.first;
        b = front.second.second;

        if (i == index)
            continue;

        if (!isSameSet(a, b)) {
            mst += front.first;
            unionSets(front.second.first, front.second.second);
        }
    }

    return sz>1 ? 1e9 : mst;
}

int main() {
        cin >> n >> e;

        for(int i=0; i<e; i++){
            cin >> src >> dest >> cost;
            edgeList.push_back(make_pair(cost, pair<int, int>(src, dest)));
        }

        initialize();

        sort(edgeList.begin(), edgeList.end());

        int mst = 0;
        for(int i=0; i<edgeList.size(); i++){
            pair<int, pair<int, int> > front = edgeList[i];
            if (!isSameSet(front.second.first, front.second.second)) {
                mst += front.first;
                unionSets(front.second.first, front.second.second);
                tree.push_back(i);
            }
        }

        if (sz>1){
            cout << "No way" << endl;
            return 0;
        }

        int res = 1e9;
        for(int i=0; i<tree.size(); i++)
```

```
            res = min(res, kruskal(tree[i]));

        if(res == 1e9)
            cout << "No Second Best MST" << endl;
        else
            cout << res << endl;

        return 0;
}
```