//kruskals mst using disjoint set

#include<bits/stdc++.h>

using std::cout;

using std::vector;

using std::pair;

//using namespace std;

//using std::iterator;

int parent[1000009];

int rank[10000009];

vector<pair<int,pair<int,int>>>edges;

vector<pair<int,pair<int,int>>>edges2;//stores the wt,src,dest of edges

int mst\_wt=0,v,e;

void addEdge(int u, int v, int w)

{

edges.push\_back({w, {u, v}});

}

class disjointset

{

public:

void makeset(int a)

{

parent[a]=a;

rank[a]=0;

}

int find(int u)

{

if (u != parent[u])

parent[u] = find(parent[u]);

return parent[u];

}

void union1(int u,int v)

{

int paru=find(u);

int parv=find(v);

if(paru==parv)

return;

if(rank[paru]>=rank[parv])

{

if(rank[paru]==rank[parv])

rank[paru]++;

parent[parv]=paru;

}

else

{parent[paru]=parv;}

}

};

int kruskalMST()

{

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

disjointset obj;

using namespace std;

for(int i=0;i<=20;i++)

{

obj.makeset(i);

}

vector< pair<int, pair<int,int> >>::iterator it;

for(it=edges.begin();it!=edges.end();it++)

{

// cout<<"\*";

int u=(\*it).second.first;

int v=(\*it).second.second;

int w=(\*it).first;

cout<<u<<v<<w<<"\n";

int setu=obj.find(u);

int setv=obj.find(v);

// cout<<setu<<setv<<"\n";

if(setu==setv)

{;}

else

{

//using namespace std;

// cout<<"\*";

edges2.push\_back({w, {u, v}});

mst\_wt += it->first;

obj.union1(u,v);

}

}

return mst\_wt;

}

int main()

{

v = 9; e = 14;

using namespace std;

vector<pair<int,pair<int,int>>>::iterator it;

addEdge(0, 1, 4);

addEdge(0, 7, 8);

addEdge(1, 2, 8);

addEdge(1, 7, 11);

addEdge(2, 3, 7);

addEdge(2, 8, 2);

addEdge(2, 5, 4);

addEdge(3, 4, 9);

addEdge(3, 5, 14);

addEdge(4, 5, 10);

addEdge(5, 6, 2);

addEdge(6, 7, 1);

addEdge(6, 8, 6);

addEdge(7, 8, 7);

int mst\_wt = kruskalMST();

/\*for(it=edges2.begin();it!=edges2.end();it++)

{

cout<<it->first<<" "<<it->second.first<<" "<<it->second.second<<"\n";

}\*/

/\*for(int i=1;i<=14;i++)

cout<<parent[i]<<"\n";\*/

cout << "\nWeight of MST is " << mst\_wt;

return 0;

}