**Kruskal using disjoint**

#include<iostream>

#include<vector>

#include<algorithm>

using namespace std;

int p[100];

int find(int x)

{

if(p[x]==x)

{

return x;

}

return find(p[x]);

}

void un(int x,int y)

{

int f\_x=find(x);

int f\_y=find(y);

p[f\_x]=f\_y;

}

int main()

{

for(int i=0;i<100;i++)

{

p[i]=i;

}

int n,m;

int a,b,w;

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

cin>>n>>m;

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

{

cin>>a>>b>>w;

edges.push\_back(make\_pair(w,make\_pair(a,b)));

}

cout<<endl;

int mst\_w=0,mst\_e=0;

int mst\_n=0;

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

while((mst\_e<n-1)||(mst\_n<m))

{

a=edges[mst\_n].second.first;

b=edges[mst\_n].second.second;

w=edges[mst\_n].first;

if(find(a)!=find(b))

{

un(a,b);

mst\_w+=w;

cout<<a<<" "<<b<<" "<<w<<endl;

mst\_e++;

}

mst\_n++;

}

cout<<"\nWeight of MST is "<<mst\_w<<endl;

return 0;

}