

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int par[205],n,m,a,cnt;
4 struct edge{
5     int u,v,w;
6 };
7 vector<edge>vv,uu;
8 bool cmp(edge x, edge y){
9     return x.w<y.w;
10 }
11 int findparent(int x){
12     if(par[x]==x) return x;
13     return par[x]=findparent(par[x]);
14 }
15 void Unionparent(int x,int y){
16     par[findparent(y)] = findparent(x);
17 }
18
19 int kruskalMST(){
20     for(int i=1; i<=n; i++) par[i]=i;
21     int sum=0,cnt=0;
22
23     sort(vv.begin(),vv.end(),cmp);
24
25     uu.clear();
26     for(int i=0; i<vv.size(); i++){
27         edge top = vv[i];
28         int parU = findparent(top.u);
29         int parV = findparent(top.v);
30
31         if(parU!=parV){
32             Unionparent(parU,parV);
33             sum += top.w;
34             cnt++;
35             edge pp; pp.u=top.u, pp.v=top.v, pp.w=top.w;
36             uu.push_back(pp);
37         }
38     }
39
40     vv.clear();
41     vv = uu;
42
43     if(cnt==n-1) return sum;
44     return -1;
45 }
46 int main(){
47     int tt; scanf("%d",&tt);
48     for(int ks=1; ks<=tt; ks++){
49         printf("Case %d:\n",ks);
50         scanf("%d%d",&n,&m);
51         vv.clear();
52
53         for(int i=1; i<=m; i++){
54             int u,v,w;
55             scanf("%d%d%d",&u,&v,&w);
56             edge pp; pp.u=u, pp.v=v, pp.w=w;
57             vv.push_back(pp);
58
59             int ans = kruskalMST();
60             printf("%d\n",ans);
61         }
62     }
63     return 0;
64 }
```