

```
1  /// Kruskal( 1174-IP-TV):
2  #include<bits/stdc++.h>
3  using namespace std;
4  int par[2005],nn;
5  map<string,int>mp;
6  struct edge{ int u,v,w; };
7  vector<edge>vv;
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 int kruskalMST(int n){
19     for(int i=1; i<=n; i++) par[i]=i;
20     int sum=0,cnt=0;
21
22     sort(vv.begin(),vv.end(),cmp);
23
24     for(int i=0; i<vv.size(); i++){
25         edge top = vv[i];
26         int parU = findparent(top.u);
27         int parV = findparent(top.v);
28
29         if(parU!=parV){
30             Unionparent(parU,parV);
31             sum += top.w;
32             cnt++;
33             if(cnt==n-1)break;
34         }
35     }
36     vv.clear();
37     return sum;
38 }
39 int main(){
40     int tt; scanf("%d",&tt);
41     for(int ks=0; ks<tt; ks++){
42         if(ks)printf("\n");
43         int n; scanf("%d",&n);
44         int m; scanf("%d",&m);
45
46         int k=0; mp.clear();
47         for(int i=1; i<=m; i++){
48             string s1,s2; int w;
49             cin >> s1 >> s2 >> w;
50
51             if(mp[s1]==0)mp[s1]=++k;
52             if(mp[s2]==0)mp[s2]=++k;
53
54             int u = mp[s1]; int v = mp[s2];
55             edge pp; pp.u=u, pp.v=v, pp.w=w;
56             vv.push_back(pp);
57         }
58
59         int ans = kruskalMST(n);
60         printf("%d\n",ans);
61     }
62     return 0;
63 }
```