

```

1. #include<bits/stdc++.h>
2. #define N 60100
3. using namespace std;
4. struct A{
5.     int u;
6.     long long v;
7.     bool operator <(const A & o) const{
8.         return v>o.v;
9.     }
10. };
11. priority_queue< A > h;
12. unordered_map<int,int> mp;
13. int p[N];
14. long long food[510],dis[N];
15. bool res[N];
16. vector<pair<int,int>> g[N];
17. int findr(int i){
18.     if(p[i]==i) return i;
19.     return p[i]=findr(p[i]);
20. }
21. int main()
22. {
23.     int q,n,m,r,k,u,v,w,idx=0;
24.     long long ans=0;
25.     scanf("%d",&q);
26.     while(q--){
27.         int idx=ans=0;
28.         scanf("%d %d %d %d",&n,&m,&r,&k);
29.         for(int i=1;i<=n;i++)p[i]=i;
30.         for(int i=1;i<n;i++){
31.             scanf("%d %d %d",&u,&v,&w);
32.             g[u].push_back({v,w});
33.             g[v].push_back({u,w});
34.         }
35.         for(int i=1;i<=r;i++){
36.             scanf("%d %d",&u,&v);
37.             p[u]=findr(p[v]);
38.             res[u]=res[v]=true;
39.         }
40.         scanf("%d",&k);
41.         for(int i=1;i<=n;i++){
42.             dis[i]=3e9;
43.             if(res[i] && !mp[findr(i)]) mp[findr(i)]=++idx;
44.         }
45.         dis[k]=0;
46.         h.push({k,0});
47.         while(!h.empty()){
48.             u=h.top().u;
49.             v=h.top().v;
50.             h.pop();
51.             if(res[u] && food[mp[findr(u)]]<dis[u]) food[mp[findr(u)]]=dis[u];
52.             for(auto x: g[u]){
53.                 if(dis[x.first]>dis[u]+x.second){
54.                     dis[x.first]=dis[u]+x.second;
55.                     h.push({x.first,dis[x.first]});
56.                 }
57.             }
58.         }
59.         for(int i=1;i<=m;i++) ans+=food[i],food[i]=0;
60.         printf("%lld\n",ans*2);
61.         for(int i=1;i<=n;i++){
62.             g[i].clear();

```

```
63.         res[i]=0;
64.     }
65.     mp.clear();
66. }
67. return 0;
68. }
```