

```

1 // 1002 - Country Roads
2 #define inf 1000000000
3 int n,m,d[505],cost[505][505]; vector<int>ed[505];
4 struct edge{
5     int u,v,w;
6     edge(int x, int y, int c){ u=x;v=y;w=c; }
7     bool operator < (const edge& p) const{ return p.w < w; }
8 };
9 void primsMST(int s){
10    priority_queue<edge>pq;
11    for(int i=0; i<ed[s].size(); i++){
12        pq.push(edge(s,ed[s][i],cost[s][ed[s][i]]));
13    }
14    memset(d,-1,sizeof(d)); d[s]=0;
15
16    while(!pq.empty()){
17        edge nd(-1,-1,-1); nd = pq.top(); pq.pop();
18        int u=nd.u, v=nd.v, w=nd.w;
19        if(d[v]!=-1) continue; // creating cycle
20        d[v] = max(d[u],w);
21        for(int i=0; i<ed[v].size(); i++){
22            if(d[ed[v][i]]==-1){
23                pq.push(edge(v,ed[v][i],cost[v][ed[v][i]]));
24            }
25        }
26    }
27}
28 int main(){
29    int tt; scanf("%d",&tt);
30    for(int ks=1; ks<=tt; ks++){
31        scanf("%d%d", &n, &m);
32        for(int i=0; i<=n; i++){
33            for(int j=0; j<=n; j++){
34                cost[i][j]=inf;
35            }
36        }
37
38        for(int i=1; i<=m; i++){
39            int u,v,w; scanf("%d%d%d", &u, &v, &w);
40            if(cost[u][v]==inf){
41                ed[u].push_back(v); ed[v].push_back(u);
42            }
43            cost[u][v]=cost[v][u]=min(cost[u][v],w);
44        }
45        int s; scanf("%d",&s);
46
47        primsMST(s);
48
49        printf("Case %d:\n",ks);
50        for(int i=0; i<n; i++){
51            if(d[i]!=-1)printf("%d\n",d[i]);
52            else printf("Impossible\n");
53        }
54        for(int i=0; i<=n; i++)ed[i].clear();
55    }
56
57 /**
58 ** Another Solution
59 This problem wants you to find the minimum cost to reach every city from a given city(my city).
60 Here minimum cost signifies the maximum road cost along the path from a city to another city,
61 you have to minimize this cost. This problem can easily be solved using modified dijkstra.
62 Set the distance of the start node(my city t) to 0 and then run dijkstra from it. Instead of
63 adding the road cost to the node in the relax function, you have to relax the current node
64 using the maximum road cost since you have to minimize the maximum road cost in the path.
65 The relax function can be written like this:
66 //here u is the source node and v is the destination node and w is the road cost from u to v
67 //and dis array stores the cost of the node upon which we'll relax
68 //if( dis[ v ] > max( w , dis[ u ] ) ) {
69 //    dis[ v ] = max( w , dis[ u ] );
70 //}
71 Time Complexity: O(E * Lg V). E is the number of roads and V is the number of cities.
72 */

```