General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
126463711	Practice: ArtLeywin	1018891 - 1	GNU C++17	Accepted	826 ms	53616 KB	2021-08-19 18:47:09	2021-08-19 18:47:09		Compare

→ Source

Copy

```
#include <bits/stdc++.h>

//#define endl '\n'
//#define int long long
#define pb push_back
#define ff first
#define ss second
#define all(x) (x).begin(), (x).end()
#define ll long long
#define ii pair<int,int>
#define vi vector<int>
#define vll vector<ll>
#define vii vector<ii>

const int INF = 0x3f3f3f3f;
const ll  LINF = 0x3f3f3f3f3f3f3f3fLL;
const int maxn = 1e5+10;
const int mod = 1e9+7;
const int LOG = 22;

using namespace std;

vector<vector<ii>> mst;
vector<vector<ii>> up;
vi deepth;

struct DSU {

        vi parent, rank;

        DSU(int N){
                parent = vi(N);
                rank = vi(N,1);
                iota(all(parent),0);
        }

        int find_set(int x){
                return parent[x] == x ? x : find_set(parent[x]);
        }

        void union_set(int a,int b){
                a = find_set(a);
                b = find_set(b);
                if(a == b) return;
                if(rank[a] > rank[b]) swap(a,b);
                parent[a] = b;
                rank[b] += rank[a];
        }

};

struct Edge{
        int u, v, w;
};

void dfs(int u, int p, int c, int d){

        deepth[u] = d;

        up[u][0] = {p, c};

        for(int i=1;i<=LOG;++i){
                int nxt = up[u][i-1].ff;
                int c = up[u][i-1].ss;
                up[u][i].ff = up[nxt][i-1].ff;
                up[u][i].ss = max(c, up[nxt][i-1].ss);
        }

        for(auto &[v, w]: mst[u]){

                if(v != p)
                        dfs(v, u, w, d+1);
```

```
        }

}

int get_max_weight(int a, int b){

        if(deepth[a] > deepth[b])
                swap(a,b);

        int d_diff = deepth[b] - deepth[a];

        int ret = 0;

        while(d_diff > 0){

                int log = log2(d_diff);
        ret = max(ret, up[b][log].ss);

        b = up[b][log].ff;

        d_diff -= (1 << log);

        }

        while (a != b) {
        int i = log2(deepth[b]);

        while (i > 0 && up[a][i].ff == up[b][i].ff)
            i--;

        ret = max(ret, up[a][i].ss);
        ret = max(ret, up[b][i].ss);

        a = up[a][i].ff;
        b = up[b][i].ff;
    }

        return ret;

}

void solve(){

        int N, R;

        cin >> N >> R;

        vector<struct Edge> graph(R);
        map<ii,int> Roads;

        for(int i=0;i<R;++i){
                cin >> graph[i].u >> graph[i].v >> graph[i].w;
                Roads[{graph[i].u,graph[i].v}] = graph[i].w;
                Roads[{graph[i].v,graph[i].u}] = graph[i].w;
        }

        sort(all(graph),
        [](struct Edge a, struct Edge other){
                if(a.w != other.w)
                        return a.w < other.w;
                if(a.u != other.u)
                        return a.u < other.u;
                return a.v < other.v;
        });

        DSU uf(N+1);

        int mst_cost = 0;

        mst = vector<vector<ii>>(N+1,vector<ii>());
        up = vector<vector<ii>>(N+1,vector<ii>(LOG+1));
        deepth = vi(N+1);

        for(auto &[u, v, w] : graph){
                if(uf.find_set(u) != uf.find_set(v)){
                        mst[u].pb({v,w});
                        mst[v].pb({u,w});
                        uf.union_set(u,v);
                        mst_cost += w;
                }
        }

        up[1][0] = {1,0};

        dfs(1, 0, 0, 0);

        int q;

        cin >> q;

        while(q--){

                int u, v;

                cin >> u >> v;
```

```
                int cost = get_max_weight(u,v);

                cout << mst_cost - cost + Roads[{u,v}] << endl;

        }

}

int32_t main(){
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        int t=1;
        //cin>>t;
        while(t--){
                solve();
        }
        return 0;
}
```
