

图论

1. 最短路

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
const int inf = 0x3f3f3f3f;
typedef pair<int,int> PII;
vector<PII> mp[100100];
int n,m,s;
int dis[100100];
int vis[100100];
priority_queue<PII,vector<PII>,greater<PII> > q;

void dj(int s)
{
    for(int i=1;i<=n;i++)dis[i]=inf;
    dis[s]=0;
    q.push({dis[s],s});

    while(!q.empty())
    {
        int u=q.top().second;
        q.pop();
        if(vis[u])continue;
        vis[u]=1;
        for(auto [w,v]:mp[u])
        {
            if(dis[u]+w<dis[v])
            {
                dis[v]=dis[u]+w;
                q.push({dis[v],v});
            }
        }
    }
}

signed main() {
    cin>>n>>m>>s;
    for(int i=1;i<=m;i++)
    {
        int u,v,w;
        cin>>u>>v>>w;
        mp[u].push_back({w,v});
    }
    dj(s);
    for(int i=1;i<=n;i++)cout<<dis[i]<<" ";
```

```

    return 0;
}

```

2. 并查集

```

int fa[100100];
void init(int n)
{
    for(int i=1;i<=n;i++)fa[i]=i;
}
int find(int x)
{
    if(fa[x]==x)
        return x;
    fa[x]=find(fa[x]);
    return find(fa[x]);
}
void merge(int x,int y)
{
    fa[find(x)]=find(y);
}

```

3. 生成树

prim

```

#include <bits/stdc++.h>
using namespace std;
const int N = 5050;
const int inf=0x3f3f;
int g[N][N],dis[200200];
bool vis[200200];
int n,m,ans,u,v,w;
void init(){
    memset(g,inf,sizeof(g));
    memset(dis,inf,sizeof(dis));
}
void addedge(int u,int v,int w)
{
    if (u != v && g[u][v] > w) g[u][v] = g[v][u] = w;
} //存图
void prim(){
    dis[1]=0;
    for(int i=1;i<=n;i++){
        int t=0;
        for(int j=1;j<=n;j++){
            if(!vis[j]&&dis[j]<dis[t])t=j;
        }
        vis[t]=1;
        ans+=dis[t];
        for(int j=1;j<=n;j++){
            if(!vis[j]&&dis[j]>g[t][j]){

```

```

        dis[j]=g[t][j];
    }
}
}

signed main() {
    cin>>n>>m;
    init();
    for(int i=1;i<=m;i++){
        cin>>u>>v>>w;
        addedge(u,v,w);
        addedge(u,v,w);//无向图
    }
    prim();
    for(int i=1;i<=n;i++){
        if(!vis[i]) {
            cout << "orz";
            return 0;
        }
    }
    cout<<ans;
    return 0;
}

```

kruskal

```

#include<bits/stdc++.h>
using namespace std;
#define int long long
int fa[200100];//并查集
struct edge{
    int u,v,w;
}e[200100],mst[200100];
int n,m,k;//图的节点数n,边数m,树的边数k
int ans;
bool cmp(edge a,edge b)
{
    return a.w<b.w;
}
void init(int n)
{
    for(int i=1;i<=n;i++)
        fa[i]=i;
}
int find(int x)
{
    if(fa[x]==x)
        return x;
    else{
        fa[x]=find(fa[x]);
    }
}

```

```

        return find(fa[x]);
    }
}
void merge(int i,int j)
{
    fa[find(i)]=find(j);
}
void kruskal(){
    for(int i=1;i<=m;i++){
        if(find(e[i].u)!=find(e[i].v)){
            k++;
            mst[i].u=e[i].u;
            mst[i].v=e[i].v;
            mst[i].w=e[i].w;
            ans+=e[i].w;
            merge(e[i].u,e[i].v);
        }
    }
}
signed main()
{
    std::ios::sync_with_stdio(false);
    std::cin.tie(0);
    cin>>n>>m;
    init(n);
    for(int i=1;i<=m;i++)cin>>e[i].u>>e[i].v>>e[i].w;
    sort(e+1,e+m+1,cmp);
    kruskal();
    if(k==n-1)cout<<ans;
    else cout<<"orz";
    return 0;
}

```

4. lca

```

#include <bits/stdc++.h>

using namespace std;
#define IOS ios::sync_with_stdio(false), cin.tie(nullptr), cout.tie(nullptr);
#define int long long
#define ull unsigned long long
#define lowbit(i) ((i) & (-i))
#define ls(p) (p << 1)
#define rs(p) (p << 1 | 1)
#define rep(i, a, b) for (int i = a; i <= b; i++)
#define per(i, a, b) for (int i = a; i >= b; i--)

typedef pair<int, int> PII;
const int mod = 1e9 + 7;
const int inf = 0x3f3f3f3f;
const int N = 5e5 + 200;

```

```

int qpow(int a, int n)
{
    int ans = 1;
    while (n)
    {
        if (n & 1)
        {
            ans = ans * a % mod;
        }
        a = a * a % mod;
        n >>= 1;
    }
    return ans;
}

// lca模板题:
int n, q, root; // n个节点,q次查询, root为根节点
vector<int> mp[N];
int lg2[N];
int dep[N];
int f[N][20];
int vis[N];
void dfs(int u, int fa = 0)
{
    if (vis[u])
        return;
    vis[u] = 1;
    dep[u] = dep[fa] + 1;
    f[u][0] = fa;
    for (int i = 1; i <= lg2[dep[u]]; i++)
    {
        f[u][i] = f[f[u][i - 1]][i - 1];
    }
    for (auto v : mp[u])
    {
        dfs(v, u);
    }
}

int lca(int a, int b)
{
    if (dep[a] > dep[b])
        swap(a, b);
    while (dep[a] != dep[b])
        b = f[b][lg2[dep[b] - dep[a]]];
    if (a == b)
        return a;
    for (int k = lg2[dep[a]]; k >= 0; k--)
    {
        if (f[a][k] != f[b][k])
        {
            a = f[a][k], b = f[b][k];
        }
    }
}

```

```

    }
    return f[a][0];
}
signed main()
{
    IOS cin >> n >> q >> root;
    for (int i = 1; i < n; i++)
    {
        int u, v;
        cin >> u >> v;
        mp[u].push_back(v);
        mp[v].push_back(u);
    }
    for (int i = 2; i <= n; i++)
    {
        lg2[i] = lg2[i / 2] + 1;
    } // 预处理log2
    dfs(root);
    while (q--)
    {
        int u, v;
        cin >> u >> v;
        cout << lca(u, v) << endl;
    }
    return 0;
}

```