[AtCoder Beginner Contest 235] [https://atcoder.jp/contests/abc235]

A

直接模拟

```
int get(int a, int b, int c)
{
    return a*100 + b* 10 + c;
}
int main()
{
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    char aa, bb, cc;
    cin >> aa >> bb >> cc;
    int a = aa - '0', b = bb-'0', c= cc-'0';
    cout << get(a, b, c) + get(b,c, a) + get(c, a ,b) << endl;
    return 0;
}
```

B

模拟直接走

```
const int MAXN = 1e5+5;
int a[MAXN];
int main()
{
    ios::sync_with_stdio(0);
```

```
    cin.tie(0);
    cout.tie(0);
    int n;
    cin >> n;
    for(int i = 0; i < n; i++) cin >> a[i];
    int cur = 0;
    for(int i = 1; i < n; i++)
    {
        if(a[i] > a[cur])
        {
            cur = i;
        }
        else
        {
            break;
        }
    }
    cout << a[cur] << endl;
    return 0;
}
```

C

开一个map记录每一个数字的位置即可

或者直接开二维vector

```
map<int ,vector<int> > mv;
int main()
{
    //FIN;
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int n, q;
    cin >> n >> q;
    for(int i = 0; i < n; i++)
    {
        int num;
        cin >> num;
```

```
        mv[num].push_back(i+1);
    }
    while(q--)
    {
        int x, k;
        cin >> x >> k;
        if(mv[x].size() < k)
        {
            cout << -1 << endl;
        }
        else
        {
            cout << mv[x][k-1] << endl;
        }
    }
    return 0;
}
```

D

bfs

```
ll a, n;
const int MAXN = 1e7+5;
int m[MAXN];
int bfs()
{
   // cout << "in" << endl;
    queue<P> q;
    q.push(P(1ll, 0));
    m[1] = 1;
    m[0] = 1;
    P now;
    while(!q.empty())
    {
        now = q.front();
        q.pop();
       // cout << now.first<< ' ' << now.second<< endl;
        if(now.first == n)
        {
```

```
            return now.second;
        }
        ll nxtf = now.first * a;
        int nxts = now.second + 1;
        if(nxtf <= 10*n && m[nxtf] == 0)
        {
            q.push(P(nxtf, nxts));
            m[nxtf] = 1;
        }
        if(now.first >= 10 && now.first % 10)
        {
            vector<int> v;
            ll _now = now.first;
            while(_now)
            {
                v.push_back(_now % 10);
                _now /= 10;
            }
            v.push_back(v[0]);
            ll sum = 0;
            for(int i = v.size()-1; i >= 1; i--)
            {
                sum = sum * 10 + v[i];
            }
            nxtf = sum;
            if(nxtf <= 10*n && m[nxtf] == 0)
            {
                q.push(P(nxtf, nxts));
                m[nxtf] = 1;
            }
        }
    }

    return -1;
}
int main()
{
    //FIN;
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    cin >> a >> n;
```

```
    int ans = bfs();
    cout << ans << endl;
    //cout << bfs(aa, nn) << endl;
    return 0;
}
```

E

首先明确，若询问的边不是环中最大的边，就可以用

方法1：离线做法，直接将询问的边放进去跑一次最小生成树，看看询问的边会不会被使用到，如果用到记录一下

方法2：在线做法，每次询问环中最大的边多长

第一种就不写了，第二种倍增lca求两点直接最长边即可

```
int n, m ,q;
const int MAXN = 2e5+5;
int fa[MAXN];
void init()
{
    for(int i = 0; i <= n; i++)
    {
        fa[i] = i;
    }
}
int find_root(int x)
{
    return x == fa[x] ? x : fa[x] = find_root(fa[x]);
}
vector<P> v[MAXN];
vector<pair<int, pair<int, int>>> edges;

int f[MAXN][20], g[MAXN][20], dep[MAXN];//倍增 father weight
void dfs(int u, int fa)
{
    for(int i = 0; i < v[u].size(); i++)
    {
        int to = v[u][i].second, w = v[u][i].first;
```

```
        if(to == fa) continue;
        f[to][0] = u;
        g[to][0] = w;
        dep[to] = dep[u] + 1;
        dfs(to, u);
    }
}

void make_lca()
{
    for(int j = 1; j <= 19; j++)
    {
        for(int i = 1; i <= n; i++)
        {
            if(f[i][j-1])
            {
                f[i][j] = f[f[i][j-1]][j-1];
                g[i][j] = max(g[i][j-1], g[f[i][j-1]][j-1]);
            }
            else
            {
                f[i][j] = 0;
            }
        }
    }
}

int lca(int x, int y)
{
    if(dep[x] > dep[y])
    {
        swap(x, y);
    }
    int tmp = dep[y] - dep[x];
    int ans = 0;
    for(int i = 0; tmp; i++, tmp >>= 1)
    {
        if(tmp & 1)
        {
            ans = max(ans, g[y][i]);
            y = f[y][i];
        }
    }
```

```
    if(x == y){
        return ans;
    }
    for(int i = 19; i >= 0; i--)
    {
        if(f[x][i] != f[y][i])
        {
            ans = max(ans, max(g[x][i], g[y][i]));
            x = f[x][i];
            y = f[y][i];
        }
    }
    return max(ans,max(g[x][0],g[y][0]));
}

int main()
{
    //FIN;
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> q;
    for(int i = 0; i < m; i++)
    {
        int a, b, c;
        cin >> a >> b >> c;
        edges.push_back(make_pair(c, make_pair(a, b)));
    }
    sort(edges.begin(), edges.end());
    init();
    int cnt = 0;
    for(int i = 0; i < m; i++)
    {
        int st = edges[i].second.first, ed =
edges[i].second.second, w = edges[i].first;
        if(find_root(st) != find_root(ed))
        {
            fa[find_root(st)] = find_root(ed);
            cnt++;
            v[st].push_back(P(w, ed));
            v[ed].push_back(P(w, st));
        }
    }
```

```
    dfs(1, 0);
    make_lca();
    while(q--)
    {
        int st, ed, w;
        cin >> st >>ed >> w;
        if(lca(st, ed) > w)
        {
            cout << "Yes" << endl;
        }
        else
        {
            cout << "No" << endl;
        }

    }
    return 0;
}
```

顺带复习了一下st和lca（洛谷模板题）

```
const int MAXN = 1e5+5;
int logn[MAXN], f[MAXN][25];
int a[MAXN];
int n, m;

void init()
{
    logn[1] = 0;
    for(int i = 2; i < MAXN; i++)
    {
        logn[i] = logn[i/2] + 1;
    }
}

void make_st()
{
    for(int j = 1; j < 25; j++)
    {
        for(int i = 1; i <= n; i++)
        {
```

```
            if(i + (1 << (j-1)) <= n)
            {
                f[i][j] = max(f[i][j-1], f[i + (1 << (j-1))][j-1]);
            }
        }
    }
}

int main()
{
    //FIN;
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    for(int i = 1; i <= n; i++)
    {
        cin >> a[i];
        f[i][0] = a[i];
    }
    init();
    make_st();
    while(m--)
    {
        int x, y;
        cin >> x >> y;
        int dis = logn[y -x +1];
        int ans = max(f[x][dis], f[y - (1 << dis) + 1][dis]);
        cout << ans << endl;
    }
    return 0;
}
```

```
const int MAXN = 5e5+5;
vector<int> v[MAXN];
int n, m, root;
int f[MAXN][25], dep[MAXN];

void dfs(int u, int fa)
{
    for(int i = 0; i < v[u].size(); i++)
```

```
    {
        int to = v[u][i];
        if(to == fa) continue;
        dep[to] = dep[u] + 1;
        f[to][0] = u;
        dfs(to, u);
    }
}


void make_lca()
{
    for(int j = 1; j < 25; j++)
    {
        for(int i = 1; i <= n; i++)
        {
            if(f[i][j-1])
            {
                f[i][j] = f[f[i][j-1]][j-1];
            }
            else
            {
                f[i][j] = 0;
            }

        }
    }
}

int lca(int x, int y)
{
    //printf("lca %d -%d \ndep %d - %d\n",x, y, dep[x], dep[y]);
    if(dep[x]  > dep[y])
    {
        swap(x, y);
    }
    int tmp = dep[y] - dep[x];
    for(int i = 0; tmp; i++, tmp >>= 1)
    {
        if(tmp & 1)
        {
            y = f[y][i];
        }
```

```
    }
    //printf("transto samedep %d - %d\n", x, y);
    if(x == y)
    {
        return x;
    }
    for(int i = 24; i >= 0; i--)
    {
        if(f[x][i] != f[y][i])
        {
            //printf("jmp fa %d fa %d - %d\n", i, f[x][i], f[y]
[i]);
            x = f[x][i];
            y = f[y][i];
        }
        //printf("transto %d - %d\n", x, y);
    }
    return f[x][0];
}

int main()
{
    //FIN;
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m >> root;
    for(int i = 0; i < n - 1; i++)
    {
        int st,ed;
        cin >> st >> ed;
        v[ed].push_back(st);
        v[st].push_back(ed);
    }
    dfs(root, 0);
    make_lca();
    while(m--)
    {
        int x, y;
        cin >> x >> y;
        cout << lca(x, y) << endl;
    }
    return 0;
```

```
}
```