Algorithm Codelet

TieWay59

October 12, 2019

Contents

1 测试一级标题 2 **3**
1.1 gym-101808-K LCA-ST.cpp 3

1 测试一级标题 2

1.1 gym-101808-K LCA-ST.cpp

```cpp
/*
 * src: https://codeforces.com/gym/101808/problem/K
 * use: LCA + ST
 *
 */

#define _debug(x) cerr<<#x<<" = "<<(x)<<endl;fflush(stdout)

#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
const int maxn = 100000 + 77;
const int inf = 1e9;
const int mo = 7901;
int rmq[maxn << 1];

struct ST {
    int mm[maxn << 1];
    int dp[maxn << 1][20];

    void init(int n) {
        mm[0] = -1;
        for (int i = 1; i <= n; ++i) {
            mm[i] = mm[i >> 1] + 1;
            dp[i][0] = i;
        }
        for (int j = 1; j <= mm[n]; ++j) {
            for (int i = 1; i + (1 << j) - 1 <= n; ++i) {
                dp[i][j] =
                        rmq[dp[i][j - 1]] <
                        rmq[dp[i + (1 << (j - 1))][j - 1]] ?
                        dp[i][j - 1] :
                        dp[i + (1 << (j - 1))][j - 1];
            }
        }
    }

    int query(int a, int b) {
        if (a > b)swap(a, b);
        int k = mm[b - a + 1];
        return rmq[dp[a][k]] <=
               rmq[dp[b - (1 << k) + 1][k]] ?
               dp[a][k] :
               dp[b - (1 << k) + 1][k];
    }
};

struct Edge {
    int to, nx;
    ll w;
```

```
};
Edge edge[maxn << 1];
int totEd, head[maxn];

int F[maxn << 1];
int P[maxn];
int cntStp;

ST st;

void init() {
    totEd = 0;
    memset(head, -1, sizeof(head));
}

void addedge(int u, int v, int w) {
    edge[totEd] = {v, head[u], w};
    head[u] = totEd++;
    edge[totEd] = {u, head[v], w};
    head[v] = totEd++;
}

int CrcA, CrcB, CrcW;
bool vis[maxn];
ll dis[maxn];

void dfs(int u, int fa, int dep) {
    F[++cntStp] = u;
    rmq[cntStp] = dep;
    P[u] = cntStp;
    vis[u] = true;

    for (int v = 0, i = head[u]; ~i; i = edge[i].nx) {
        v = edge[i].to;

        if (vis[v]) {
            if (v != fa) {
                CrcA = u;
                CrcB = v;
                CrcW = edge[i].w;
            }
            continue;
        }
//        _debug(v);
        dis[v] = dis[u] + edge[i].w;
        dfs(v, u, dep + 1);
        F[++cntStp] = u;
        rmq[cntStp] = dep;
    }
}

void LCA_init(int root, int node_num) {
    cntStp = 0;
    memset(vis, 0, sizeof vis);
    memset(dis, 0, sizeof dis);
    dfs(root, -1, 0);
```

```
    st.init(2 * node_num - 1);
}

inline int LCA_query(int u, int v) {
    return F[st.query(P[u], P[v])];
}

inline ll qryDis(int u, int v) {
    return dis[u] + dis[v] - 2 * dis[LCA_query(u, v)];
}

int main() {
//    ios::sync_with_stdio(false);
//    cin.tie(nullptr);
//    cout.tie(nullptr);

    int Kase = 0, N = 0, Q = 0, root = 1, u = 0, v = 0;
//    cin >> Kase;
    scanf("%d", &Kase);
    while (Kase--) {
        scanf("%d %d", &N, &Q);
        init();
        for (int i = 1, w; i <= N; ++i) {
            scanf("%d %d %d", &u, &v, &w);
            addedge(u, v, w);
        }
        LCA_init(root, N);
        while (Q--) {
            scanf("%d%d", &u, &v);
            ll res = qryDis(u, v);
            res = min(res, qryDis(u, CrcA) + qryDis(v, CrcB) + CrcW);
            res = min(res, qryDis(v, CrcA) + qryDis(u, CrcB) + CrcW);
            printf("%lld\n", res);
        }

    }
    return 0;
}


/*
3
 3
1 2
 2 222
 3 22

 */
```