

2023“钉耙编程”中国大学生算法设计超级联赛(1)

整体过题记录 AC (5/12)

1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012
4/221	1/27	-3		1/39	-4			2/4			3/72

比赛链接: <https://acm.hdu.edu.cn/contest/problems?cid=1094>

题目分析及错误反思

1001. Hide-And-Seek Game

题型: 传统题 exgcd+lca

如果对于一个点 v 都在两个链上, 且两条链的长度为 len_1, len_2 , 到两个起点的长度为 d_1, d_2 。

所以可以列出四个方程:

- $2len_1x + d_1 = 2len_2y + d_2$
- $2len_1x - d_1 = 2len_2y + d_2$
- $2len_1x - d_1 = 2len_2y - d_2$
- $2len_1x + d_1 = 2len_2y - d_2$

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

using namespace std;

template <typename T>
void read(T& x) {
    x = 0; char ch = 0; int f = 1;
    for (; !isdigit(ch); ch = getchar()) if (ch == '-') f = -1;
    for (; isdigit(ch); ch = getchar()) x = x * 10 + (ch & 15);
    x *= f;
}

using i64 = long long;

i64 exgcd(i64 a, i64 b, i64& x, i64& y) {
    if (b == 0) {
        x = 1, y = 0;
        return a;
    }
    i64 g = exgcd(b, a % b, y, x);
    y -= (a / b) * x;
    return g;
}
```

```

i64 gcd(i64 x, i64 y) {
    return !y ? x : gcd(y, x % y);
}

void solve();

const int N = 3e3 + 5, LG = 17;

int LOG2[N << 1];

int main() {
    LOG2[0] = -1;
    for (int i = 1; i < N * 2; i++) {
        LOG2[i] = LOG2[i >> 1] + 1;
    }
    int t;
    read(t);
    while (t--) {
        solve();
    }
    return 0;
}

int n, m;
vector<int> adj[N];
int flca[N << 1][LG + 2];
int dep[N];
int dfn[N];
int par[N];
int dfc;

void dfs0(int u, int fa) {
    dep[u] = dep[fa] + 1;
    par[u] = fa;
    flca[dfn[u] = ++dfc][0] = u;
    for (int i = 0; i < int(adj[u].size()); i++) {
        int v = adj[u][i];
        if (v == fa) continue;
        dfs0(v, u);
        flca[++dfc][0] = u;
    }
}

void prework() {
    for (int j = 1; j <= LG; j++) {
        for (int i = 1; i + (1 << j) <= dfc; i++) {
            if (dep[flca[i][j - 1]] < dep[flca[i + (1 << j) - 1][j - 1]]) {
                flca[i][j] = flca[i][j - 1];
            } else {
                flca[i][j] = flca[i + (1 << j) - 1][j - 1];
            }
        }
    }
}

```

```

int lca(int u, int v) {
    if (dfn[u] > dfn[v]) swap(u, v);
    int k = LOG2[dfn[v] - dfn[u] + 1];
    int p1 = flca[dfn[u]][k];
    int p2 = flca[dfn[v] - (1 << k) + 1][k];
    return dep[p1] < dep[p2] ? p1 : p2;
}

int dist(int u, int v) {
    int LCA = lca(u, v);
    return dep[u] + dep[v] - 2 * dep[LCA];
}

bool check(int u, int v, int x) {
    return dist(u, x) + dist(v, x) == dist(u, v);
}

i64 calc(i64 len1, i64 d1, i64 len2, i64 d2) {
    i64 res = 1e18;
    i64 A = 211 * len1, B = 211 * len2;
    i64 x, y;
    i64 g = exgcd(-A, B, x, y);
    i64 aa = A / gcd(A, B), bb = B / gcd(A, B);

    i64 C = d1 - d2;
    if (C % g == 0) {
        i64 xx = C / g * x;
        xx = (xx % bb + bb) % bb;
        if (A * xx + d1 - d2 >= 0) {
            res = min(res, A * xx + d1);
        } else {
            i64 yy = C / g * y;
            yy = (yy % aa + aa) % aa;
            res = min(res, B * yy + d2);
        }
    }
    // cerr << res << "\n";
}

C = -d1 - d2;
if (C % g == 0) {
    i64 xx = C / g * x;
    xx = (xx % bb + bb) % bb;
    if (xx == 0) {
        xx = xx + bb;
    }
    // cerr << "xx = " << xx << "\n";
    if (A * xx - d1 - d2 >= 0) {
        res = min(res, A * xx - d1);
    }
    // cerr << "ard 111111 = " << res << "\n";
} else {
    i64 yy = C / g * y;
    yy = (yy % aa + aa) % aa;
    res = min(res, B * yy + d2);
    // cerr << "ard 222222 = " << res << "\n";
}

```

```

//      cerr << res << "\n";
    }

    c = d1 + d2;
    if (c % g == 0) {
        i64 xx = c / g * x;
        xx = (xx % bb + bb) % bb;
        if (A * xx + d1 + d2 > 0) {
            res = min(res, A * xx + d1);
        } else {
            i64 yy = c / g * y;
            yy = (yy % aa + aa) % aa;
            if (yy == 0) {
                yy += aa;
            }
            res = min(res, B * yy - d2);
        }
    }
    //      cerr << res << "\n";
}

    c = -d1 + d2;
    if (c % g == 0) {
//      cerr << "x = " << x << "\n";
        i64 xx = c / g * x;
        xx = (xx % bb + bb) % bb;
//      cerr << "xx = " << xx << "\n";
        if (xx == 0) {
            xx = xx + bb;
        }
        if (A * xx - d1 + d2 > 0) {
            res = min(res, A * xx - d1);
//      cerr << "xx = " << xx << "ard 111111\n";
        } else {
            i64 yy = c / g * y;
            yy = (yy % aa + aa) % aa;
            if (yy == 0) {
                yy += aa;
            }
            res = min(res, B * yy - d2);
//      cerr << "yy = " << yy << "ard 22222222\n";
        }
    }
}

    return res;
}

void solve() {
    read(n), read(m);
    for (int i = 1; i < n; i++) {
        int u, v;
        read(u), read(v);
        adj[u].push_back(v);
        adj[v].push_back(u);
    }
    dfc = 0;
}

```

```

dep[0] = 0;
dfs0(1, 0);
prework();

while (m--) {
    int s1, t1, s2, t2;
    read(s1), read(t1), read(s2), read(t2);
    int len1 = dist(s1, t1), len2 = dist(s2, t2);

    int LCA = lca(s1, t1);
    int u = s1, v = t1;
    i64 ans = 1e18;
    int g = -1;
    while (u != LCA) {
        if (check(s2, t2, u)) {
            i64 d1 = dist(u, s1), d2 = dist(u, s2);
            i64 val = calc(len1, d1, len2, d2);
            if (val < ans) {
                g = u;
                ans = val;
            }
        }
        u = par[u];
    }
    while (v != LCA) {
        if (check(s2, t2, v)) {
            i64 d1 = dist(v, s1), d2 = dist(v, s2);
            i64 val = calc(len1, d1, len2, d2);
            if (val < ans) {
                g = v;
                ans = val;
            }
        }
        v = par[v];
    }
    if (check(s2, t2, LCA)) {
        i64 d1 = dist(LCA, s1), d2 = dist(LCA, s2);
        i64 val = calc(len1, d1, len2, d2);
        if (val < ans) {
            g = LCA;
            ans = val;
        }
    }

    printf("%d\n", g);
}

for (int i = 1; i <= n; i++) {
    adj[i].clear();
    dep[i] = par[i] = dfn[i] = 0;
}

for (int i = 1; i <= n; i++) {
    for (int j = 0; j <= LG; j++) {
        flca[i][j] = 0;
    }
}

```

```
}
```

1003. Mr. Liang play Card Game

题型：传统题 区间 DP

错误原因：比赛最后时刻，多组数据没有初始化，但是评测机最后没有给出反应，所以就没有找到主要的问题。

1005. Cyclically Isomorphic

题型：字符串的最小表示法

对于每个字符串，就用最小表示法后记录哈希，就可以直接 $O(1)$ 得到答案了。

1010. Easy problem I

题型：传统题，数据结构

因为 x_j 是递增的，所以只要翻转过一次，接下来的所有情况都是要翻转的，所以其实比赛的最后一个小时就已经有正解的思路了，

赛后补题安排

题目分配

- H:
- C: 03、10
- Z: 06、10、11

题目记录

暴露问题及需要补的知识点

暴露的问题

- 比赛的时候把自己的思路整理好，然后再上机敲，整理好自己的情绪，可以紧张一点，但不要过于自信。**小张不要说：过了！秒了！**
- 沉着冷静，不要紧张！
- 队伍最后一个小时听黄陈的安排，不要像小张这样想要敲 **1010**
- 另外一个问题和上一场牛客的多校相同，需要在这个暑假通过比赛、VP 和个人训练迅速调整。
- 多组数据的初始化问题！！

需要补的知识点

- 数据结构：李超线段树
- DP：区间 DP 较难的题型
- 张的数学题！！！啊啊啊！！