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 $d1, d2$。

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

```cpp
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];
            }
        }
    }
}
```

```cpp
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 = 2ll * len1, B = 2ll * 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 1111111 = " << res << "\n";
        } else {
            i64 yy = C / g * y;
            yy = (yy % aa + aa) % aa;
            res = min(res, B * yy + d2);
//          cerr << "ard 2222222 = " << 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

1010. Easy problem I

赛中已经知道了转移的次数一定是有限的，但是并没有想好具体的实现方式，补题用了两个不同类型的线段树。

```
#include <bits/stdc++.h>
using namespace std;
#define inf (0x3f3f3f3f)

struct IO {
    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;
    }
    template <typename T>
    void write(T x) {
        if (x < 0) putchar('-'), x = -x;
        if (x > 9) write(x / 10);
        putchar(x % 10 + '0');
    }

    template <typename T>
    void writeln(T x) {
        write(x), putchar('\n');
    }
} io;

void solve();

int main() {
    int t;
    io.read(t);
    for (int i = 1; i <= t; ++i) solve();
    return 0;
}

typedef long long i64;

const int N = 2e5 + 5;

int a[N];
int n, m;

struct segment2 {  // 构造一个区间加和区间乘区间查询的线段树
    int cnt[N << 2], mul[N << 2], add[N << 2];
    i64 sum[N << 2];

    void pushup(int p) {
        cnt[p] = cnt[p << 1] + cnt[p << 1 | 1];
        sum[p] = sum[p << 1] + sum[p << 1 | 1];
    }
    void build(int p, int l, int r) {
        cnt[p] = add[p] = sum[p] = 0;
        mul[p] = 1;
        if (l == r) {
            return;
        }
        int mid = (l + r) >> 1;
        build(p << 1, l, mid);
        build(p << 1 | 1, mid + 1, r);
    }
    void apply0(int p, int v) {  // 处理乘法懒标记
        sum[p] *= v;
        add[p] *= v;
        mul[p] *= v;
    }
    void apply1(int p, int v) {  // 处理加法懒标记
        sum[p] += i64(cnt[p]) * v;
        add[p] += v;
```

```cpp
    }
    void pushdown(int p) {
        if (mul[p] != 1) {
            apply0(p << 1, mul[p]);
            apply0(p << 1 | 1, mul[p]);
            mul[p] = 1;
        }
        if (add[p] != 0) {
            apply1(p << 1, add[p]);
            apply1(p << 1 | 1, add[p]);
            add[p] = 0;
        }
    }
    void secmul(int p, int l, int r, int ql, int qr, int v) {
        if (ql <= l && r <= qr) {
            apply0(p, v);
            return;
        }
        pushdown(p);
        int mid = (l + r) >> 1;
        if (ql <= mid) secmul(p << 1, l, mid, ql, qr, v);
        if (qr > mid) secmul(p << 1 | 1, mid + 1, r, ql, qr, v);
        pushup(p);
    }
    void secadd(int p, int l, int r, int ql, int qr, int v) {
        if (ql <= l && r <= qr) {
            apply1(p, v);
            return;
        }
        pushdown(p);
        int mid = (l + r) >> 1;
        if (ql <= mid) secadd(p << 1, l, mid, ql, qr, v);
        if (qr > mid) secadd(p << 1 | 1, mid + 1, r, ql, qr, v);
        pushup(p);
    }
    void insert(int p, int l, int r, int pos, int v) {
        if (l == r) {
            cnt[p] = 1;
            sum[p] = v;
            return;
        }
        pushdown(p);
        int mid = (l + r) >> 1;
        if (pos <= mid) {
            insert(p << 1, l, mid, pos, v);
        } else {
            insert(p << 1 | 1, mid + 1, r, pos, v);
        }
        pushup(p);
    }
    i64 query(int p, int l, int r, int ql, int qr) {
        if (ql <= l && r <= qr) return sum[p];
        pushdown(p);
        i64 sum = 0;
        int mid = (l + r) >> 1;
        if (ql <= mid) sum += query(p << 1, l, mid, ql, qr);
        if (qr > mid) sum += query(p << 1 | 1, mid + 1, r, ql, qr);
        return sum;
    }
} sgt2;

struct segment1 {
    int mi[N << 2];
    i64 sum[N << 2];
    int cnt[N << 2];
    int lazy[N << 2];

    void pushup(int p) {
        sum[p] = sum[p << 1] + sum[p << 1 | 1];
        mi[p] = min(mi[p << 1], mi[p << 1 | 1]);
        cnt[p] = cnt[p << 1] + cnt[p << 1 | 1];
    }
    void apply(int p, int v) {
        mi[p] -= v;
        sum[p] -= i64(cnt[p]) * v;
        lazy[p] += v;
    }
    void pushdown(int p) {
        if (lazy[p] != 0) {
            if (cnt[p << 1] > 0) apply(p << 1, lazy[p]);
            if (cnt[p << 1 | 1] > 0) apply(p << 1 | 1, lazy[p]);
            lazy[p] = 0;
        }
    }
    void build(int p, int l, int r) {
        lazy[p] = 0;
        if (l == r) {
            mi[p] = sum[p] = a[l];
            cnt[p] = 1;
            return;
        }
        int mid = (l + r) >> 1;
        build(p << 1, l, mid);
        build(p << 1 | 1, mid + 1, r);
        pushup(p);
    }
    void sub(int p, int l, int r, int ql, int qr, int v) {
        if (ql <= l && r <= qr) {  // 已经到了修改的区间内
            if (mi[p] >= 1e7) return;
            if (mi[p] >= v) {
                apply(p, v);
            } else {
                if (l == r) {  // 插入到另外一个线段树中
                    sgt2.insert(1, 1, n, l, v - sum[p]);
                    sum[p] = cnt[p] = 0;
                    mi[p] = inf;
                } else {
                    int mid = (l + r) >> 1;
                    pushdown(p);
                    sub(p << 1, l, mid, ql, qr, v);
                    sub(p << 1 | 1, mid + 1, r, ql, qr, v);
                    pushup(p);
                }
            }
            return;
        }
        pushdown(p);
        int mid = (l + r) >> 1;
        if (ql <= mid) sub(p << 1, l, mid, ql, qr, v);
        if (qr > mid) sub(p << 1 | 1, mid + 1, r, ql, qr, v);
        pushup(p);
    }
    i64 query(int p, int l, int r, int ql, int qr) {
        if (ql <= l && r <= qr) return sum[p];
        pushdown(p);
        i64 sum = 0;
        int mid = (l + r) >> 1;
        if (ql <= mid) sum += query(p << 1, l, mid, ql, qr);
        if (qr > mid) sum += query(p << 1 | 1, mid + 1, r, ql, qr);
        return sum;
    }
} sgt1;

void solve() {
    io.read(n), io.read(m);
    for (int i = 1; i <= n; i++) io.read(a[i]);
    sgt1.build(1, 1, n), sgt2.build(1, 1, n);
    int opt, l, r, x;
    while (m--) {
        io.read(opt), io.read(l), io.read(r);
        if (opt == 1) {
            io.read(x);
            sgt2.secmul(1, 1, n, l, r, -1);
            sgt2.secadd(1, 1, n, l, r, x);
            sgt1.sub(1, 1, n, l, r, x);
        } else {
            i64 ans1 = sgt1.query(1, 1, n, l, r);
            i64 ans2 = sgt2.query(1, 1, n, l, r);
            io.writeln(ans1 + ans2);
        }
    }
}
```

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，较为困难的线段树相关的数据结构题
- DP：区间 DP 较难的题型
- 张的数学题！！！啊啊啊！！！