

TUGrader Contest 1 (Real!)

Get up!

(1 sec, 256 MB)

Hint 1 เราไม่จำเป็นต้องเริ่มที่ s กล่าวคือ ไม่ว่าใครจะเป็นคนปลูกใคร ลำดับที่เท่าไร ย่อมให้ผลเหมือนกัน

Hint 2 ปัญหาดังกล่าวเป็นปัญหาที่ดัดแปลงจากปัญหาคลาสสิก นั่นคือ ต้นไม้แผ่ทั่วน้อยสุด

Solution

สังเกตได้ว่า ต้องการหา Minimum Spanning Tree บน Complete Graph ซึ่งมีอัลกอริทึมที่ใช้ทั่วไป 2 ตัว คือ Kruskal's Algorithm และ Prim's Algorithm แต่เนื่องจากกราฟหนามาก จึงไม่สามารถใช้ Kruskal Algorithm ได้ เพราะจะใช้เวลาเป็น $O(N^2)$ (ประมาณ 10000 MB ซึ่งไม่เพียงพอสำหรับทรัพยากรที่มีอยู่) เมื่อใช้ Prim's Algorithm จะให้ความจำพอดี แต่จะใช้เวลาเกินไถ่ นั่นคือ $O(N^2 \log N)$ เราจึงต้องดัดแปลงขั้นตอนวิธีการ ให้ใช้เวลาเพียง $O(N^2)$ โดยเราจะทำดังนี้

1. เลือกโหนด c ที่ยังไม่อยู่ใน MST ที่ cost จาก MST ไปยัง c มีค่าน้อยสุด
2. Update ค่า cost ของทุกโหนดรอบ c ให้มีค่าเป็น min ของค่าปัจจุบันกับ cost จาก c ไปยังโหนดนั้น
3. นำ c เข้า MST
4. ทำซ้ำขั้นตอนที่ 1 จนกว่าทุกโหนดจะอยู่ใน MST

สำหรับ Subtask 1 ต้องการให้เดินมั่วเท่านั้น ($O(N!)$ ก็ผ่าน)

Subtask 2 ต้องการให้ใช้ Kruskal's Algorithm

Subtask 3 ต้องการให้ใช้ Prim's Algorithm

Subtask 4 ต้องการให้ใช้ Prim's Algorithm แบบดัดแปลง

Number

(1 sec, 64 MB)

Hint 1: Greedy

Hint 2: Sort

Solution

ข้อนี้ไม่มีอะไรยาก สังเกตว่าผลต่างจะน้อยสุดได้เราต้องหาผลต่างของตัวปัจจุบันกับตัวที่น้อยที่สุดที่มากกว่าหรือเท่ากับปัจจุบัน แล้วหาค่า min ระหว่างค่าพวกนี้ เราจึงทำการเรียงข้อมูลก่อนแล้วค่อยๆไล่ไปเรื่อยๆ นั่นคือสำหรับทุกๆค่าที่ไม่ใช่ค่ามากที่สุด ให้หาค่าต่ำสุดระหว่างผลต่างของแต่ละตัวกับตัวที่น้อยที่สุดที่มากกว่ามัน

พูดง่ายๆคือ

```
mn = infinity
for i in range(N-1):
    mn = min(mn,A[i+1] - A[i])
return mn
```

Red Node

(1.5 sec, 256 MB)

ข้อนี้มีวิธีแก้หลักๆ 3 วิธีคือ

1. Centroid Decomposition of a Tree
2. Heavy-Light Decomposition
3. Square-Root Optimization

โดยในที่นี้จะละ solution 1 และ 2 ไว้เนื่องจากซับซ้อนเกินไป (เกินขอบเขตเนื้อหา)

หากสนใจ solution 1, 2 สามารถติดต่อคุณ aquablitz11 ได้

Square-Root Optimization ในที่นี้คือค่อยๆ ทำทีละ \sqrt{Q} queries โดยแล้วค่อย update ที่เดียวหลังจบทั้ง \sqrt{Q} queries นั่นคือ แต่ละคำสั่ง หากเป็นคำสั่ง ระบายสี ก็ เก็บใส่ใน container ชั่วคราว แต่หากเป็นคำสั่ง ถูม ก็ค่อยๆไล่ใน container ชั่วคราว $O(\sqrt{Q})$ กับของเดิมที่มีอยู่แล้ว $O(\log N)$ โดยรวมแล้วจะใช้เวลาไม่เกินอันดับรากที่สอง เมื่อทำจบแล้วก็ update ที่เดียวก็ใช้เวลา $O(N)$ ทำไปเรื่อยๆทั้งหมด $O(\sqrt{Q})$ รอบ ใช้เวลารวม $O(N(\sqrt{Q} + \log N))$

ส่วน Subtask แรกต้องการเพียงแค่ Breadth First Search ธรรมดา

ปล. จากการแข่งขันพบว่าหลายคนใช้ Breadth First Search + Lazy Propagation + Pruning จนทำให้ Run time ออกมาดีกว่า Official Solution ด้วยซ้ำ จึงจำเป็นต้องมีการเพิ่มเทสเคส ขอร้องกับข้อผิดพลาดในครั้งนี้นี้ด้วยครับ

Implementation Code (C++)

Get up!

```
#include <bits/stdc++.h>
using namespace std;
const int N = 15010;
typedef long long ll;
typedef pair<ll,ll> pll;
const ll INF = 1e18;

int n, p, s, o;
int x[N], y[N], ix[N];
ll v[N];

ll dist(int x1, int y1, int x2, int y2) {
    ll dx = abs(x1-x2);
    ll dy = abs(y1-y2);
    if (p == 2) dx = dx*dx, dy = dy*dy;
    else if (p == 3) dx = dx*dx*dx, dy = dy*dy*dy;
    return dx+dy;
}
ll dist(int i1, int i2) { return dist(x[i1], y[i1], x[i2], y[i2]); }

int main()
{
    scanf("%d%d%d%d", &n, &p, &s, &o);
    for(int i = 0; i < n; i++)
        scanf("%d%d",&x[i], &y[i]);
    ll sum = 0;
    for(int i = 0; i < n; i++) v[i] = INF, ix[i] = i;
    v[s-1] = 0;
    for(int i = 0; i < n; i++) {
        pll mn(INF, 0);
        for(int j = 0; j < n-i; j++) mn = min(mn, pll(v[ix[j]], j));
        sum += mn.first;
        swap(ix[mn.second], ix[n-i-1]);
        for(int j = 0; j < n-i-1; j++)
            v[ix[j]] = min(v[ix[j]], 2*dist(ix[n-i-1], ix[j]) + o);
    }
    printf("%lld",sum);

    return 0;
}
```

Solution By: aquablitz11

Number

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n; scanf("%d", &n);
    vector<long long> V;
    for(int i = 0; i < n; ++i) {
        long long ret; scanf("%lld", &ret);
        V.emplace_back(ret);
    }
    sort(V.begin(), V.end());
    long long mn = 1e18+711;
    for(int i = 0; i < V.size()-1; ++i) mn = min(mn, V[i+1] - V[i]);
    cout << mn;
}
```

Solution By: win11905

Red Node

```
#include <bits/stdc++.h>
using namespace std;

const int MAXN = 1<<18, inf = 1e9;

int n, m, dep[MAXN], par[MAXN][18], d[MAXN];
vector<int> g[MAXN];

void dfs(int u, int p) {
    dep[u] = dep[p] + 1;
    par[u][0] = p;
    for(int i = 1; i <= 17; ++i) par[u][i] = par[par[u][i-1]][i-1];
    for(auto v:g[u]) if(v != p) dfs(v, u);
}

int lca(int a, int b) {
    if(dep[a] < dep[b]) swap(a, b);
    for(int i = 17; i >= 0; --i) if(dep[par[a][i]] >= dep[b]) a = par[a][i];
    if(a == b) return a;
    for(int i = 17; i >= 0; --i) if(par[a][i] != par[b][i]) a = par[a][i], b = par[b][i];
    return par[a][0];
}

int main() {
    fill(d, d+MAXN, inf);
    scanf("%d %d", &n, &m);
    for(int i = 1; i < n; ++i) {
        int a, b;
        scanf("%d %d", &a, &b);
        g[a].emplace_back(b);
        g[b].emplace_back(a);
    }
    dfs(1, 0);
    vector<int> ret;
    ret.emplace_back(1);
    for(int i = 1; i <= m; ++i) {
        int a, b;
        scanf("%d %d", &a, &b);
        if(a == 1) ret.emplace_back(b);
        else {
            int mn = d[b];
            for(auto v:ret) mn = min(mn, dep[v] + dep[b] - 2*dep[lca(v, b)]);
            printf("%d\n", mn);
        }
        if(i % 317 == 0) {
            queue<int> Q;
            for(auto v:ret) {
                d[v] = 0;
                Q.push(v);
            }
            ret.clear();
            while(!Q.empty()) {
                auto u = Q.front();
                Q.pop();
                for(auto v:g[u]) {
                    if(d[v] > d[u] + 1) {
                        d[v] = d[u] + 1;
                        Q.push(v);
                    }
                }
            }
        }
    }
}
```

Solution By: win11905

--END--

Thanks for your participation.

Thank you

Asst. Prof. Jittat Fakcharoenphol, Ph. D

Asst. Prof. Nattee Niparnan, Ph. D

For the Prototypic Cafe Grader System.

Credits

win11905 - Grader Admin, Problemsetter

aquablitz11 - Grader Manager, Problemsetter

1048576 - Grader Manager, Problemsetter

Phon1209 - Grader Manager

Good luck.