Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0241 – Кудашин Олег Кириллович

A	B	C	D	E	F	sum
30	100	100	100	40	.	370

Task A

```
// T1 OSSPBU

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef pair<ll, ll> pll;

const ll MAXN = 100;

ll BC[MAXN][MAXN];

set<pll> dio(ll a, ll b, ll s) {
    set<pll> ans;
    for (ll x = 0; x < 100; ++x)
        for (ll y = 0; y < 100; ++y)
            if (x * a + y * b == s)
                ans.insert({x, y});
    return ans;
}

vector<ll> divs(ll x) {
    vector<ll> d;
    for (ll i = 1; i * i <= x; ++i) {
        if (x % i == 0) {
            d.push_back(i);
            if (i * i != x)
                d.push_back(x / i);
```

```
        }
    }
    sort(d.begin(), d.end());
    return d;
}

ll C(ll n, ll k) {
    return BC[n - k][k];
}

vector<ll> pr;
ll a, b, n;

ll binCnt(ll n, ll x) {
    vector<vector<ll>> dp(MAXN, vector<ll>(MAXN));
    for (int len = 1; len <= n; ++len) {
        for (int x = 0; x <= len; ++x) {
            dp[len][x] += C(len, x);

            for (int j = 2; len * j < MAXN; ++j)
                dp[len * j][x * j] -= dp[len][x];
        }
    }
    ll per = gcd(n, x);
    auto d = divs(per);
    ll res = 0;
    if (per != n)
        dp[n][x];
    for (int len = 1; len <= n; ++len)
        for (int y = 0; y <= len; ++y) {
            if (n % len == 0 && (y * n) / len == x)
                res += dp[len][y] * len * (a * y + b * (len - y));
        }

    return res;
}

void solve() {
    cin >> a >> b >> n;
    auto dc = dio(a, b, n);
    if (dc.empty()) {
        cout << 0 << '\n';
        return;
    }

    vector<ll> pc(100);
    auto d = divs(n);
    vector<set<pll>> er;

    ll res = 0;

    for (auto i : dc) {
        ll x = i.first, y = i.second;
```

```
        ll m = x + y;
        res += binCnt(m, x);
    }
    cout << res << '\n';
}

int main() {
    BC[0][0] = 1;
    for (int i = 0; i < MAXN; ++i)
        BC[0][i] = BC[i][0] = 1;

    for (ll sum = 1; sum < MAXN; ++sum)
        for (ll x = 1; x < sum; ++x)
            BC[sum - x][x] = BC[sum - x][x - 1] + BC[sum - x - 1][x];

    bitset<MAXN * 3> isp;
    isp.flip();
    isp[0] = isp[1] = 0;
    for (int i = 2; i < MAXN; ++i) {
        if (isp[i]) {
            pr.push_back(i);
            for (int j = 2; i * j < MAXN; ++j)
                isp[i * j] = false;
        }
    }

    ios_base::sync_with_stdio(0);
    cout.tie(0);
    solve();
    return 0;
}
```

Task B

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

using namespace std;

typedef long long ll;
typedef pair<ll, ll> pll;

const ll MAXN = 100;

struct pt {
    ll x = 0, y = 0;

    pt() = default;

    pt(ll X, ll Y) {
        x = X, y = Y;
    }
};
```

```
pt operator+(pt A, pt B) {
    A.x += B.x;
    A.y += B.y;
    return A;
}

ll hexDist(pt A, pt B) {
    ll dx = B.x - A.x, dy = B.y - A.y;
    if ((dx >= 0 && dy <= 0) || (dx <= 0 && dy >= 0))
        return max(abs(dx), abs(dy));
    return abs(dx) + abs(dy);
}

void solve() {
    ll n, tt;
    cin >> n;
    pt A, B;
    cin >> A.x >> A.y >> tt >> B.x >> B.y >> tt;
    ll dist = hexDist(A, B);
    vector<pt> mv = {{-n,      -n},
                     {n,       n},
                     {-2 * n, n},
                     {2 * n,  -n},
                     {n,       -2 * n},
                     {-n,      2 * n}};

    int m = 6;
    for (int i = 0; i < (1 << 6); ++i) {
        pt C = A;
        for (int j = 0; j < m; ++j) {
            if ((i >> j) & 1)
                C = C + mv[j];
        }
        dist = min(dist, hexDist(B, C));
    }
    cout << dist << '\n';
}

int main() {

    ios_base::sync_with_stdio(0);
    cout.tie(0);
    solve();
    return 0;
}
```

Task C

```
// guess >> str

#include <bits/stdc++.h>

using namespace std;
```

```
typedef long long ll;
typedef pair<ll, ll> pll;

const ll MAXN = 100;

vector<ll> MODS = {1000000007, 440739714, 625754008, 907667358,
    905606960, 553951302, 222645129, 86480776,
                    54559835,
                    49444567, 116156981, 209649375, 207609493, 182128177,
    2449289};


mt19937 rnd(666);

const ll GLM = 1e9 + 7;

ll MOD(ll x, ll M) {
    if (x < 0)
        x += M;
    if (x >= GLM)
        x %= GLM;
    return x;
}


ll strHash(string s, ll M) {
    ll H = 0;
    for (auto i : s) {
        H = H * 2 + int(i - '0');
        if (H >= M)
            H -= M;
    }
    return H;
}

ll hAdv(ll H, ll c, ll bin, ll M) {
    H = MOD(H - bin * c, M);
    return (H * 2 + c) % M;
}


string rot(string s, ll x) { // start goes to x
    string r = s;
    int n = s.size();
    for (int i = 0; i < n; ++i)
        r[i] = s[(i + x) % n];
    return r;
}

string toSys(ll x, ll grp = 1) {
    string t(30, '0');
    for (int i = 0; i < 30; ++i)
```

```
        t[i] = char('0' + ((x >> i) & 1));
    if (grp == 1)
        return t;
    string r((30 / grp), '0');
    for (int i = 0; i < (30 / grp); ++i) {
        int val = 0;
        for (int j = 0; j < grp; ++j)
            val = val * 2 + (t[i * grp + j] - '0');
        r[i] = char(32 + val);
    }
    return r;
}

ll rHash(string s, ll M) {
    ll n = s.size();
    ll bin = 1;
    for (int i = 1; i < n; ++i)
        bin = (bin * 2) % M;

    ll H = strHash(s, M);
    ll HM = H;
    for (int i = 0; i < n - 1; ++i) {
        H = hAdv(H, s[i] - '0', bin, M);
        HM = (HM * H) % M;
    }
    return HM;
}

int GRP = 6;

string fullHash(string s) {
    string H;
    for (auto CM : MODS)
        H += toSys(rHash(s, CM), GRP);
    return H;
}

void test() {
    string s;
    int n = 100000;
    for (int i = 0; i < n; ++i)
        s += char('0' + (rnd() & 1));
    string HA = fullHash(s);

    for (int j = 0; j < n; ++j) {
        string t = rot(s, j);
        string HB = fullHash(t);
        if (HB != HA)
            cout << "WT! " << j << endl;
    }
}

void solve() {
```

```
    //test();
    string s, ty;
    int n;
    cin >> ty >> n;
    getline(cin, s);
    getline(cin, s);

    if (ty == "first") {
        string H = fullHash(s);
        cout << H << '\n';
    } else {
        string t;
        getline(cin, t);

        string H = fullHash(s);
        t.resize(H.size(), ' ');
        //cout << t.size() << endl;
        //cout << H.size() << endl;
        if (H == t)
            cout << "YES\n";
        else
            cout << "NO\n";
    }
}

int main() {
//    for (int i = 0; i < 20; ++i)
//        cout << rnd() % ll(1e9 + 7) << ", ";
//    cout <<
    ios_base::sync_with_stdio(0);
    cout.tie(0);
    solve();
    return 0;
}
```

Task D

```
// guess >> str

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef pair<ll, ll> pll;

const ll MAXN = 100;

int ptr = 0;
int n;

bool lastAns = 0;
```

```
bool ask(int i, int j) { // is <?
    cout << i + 1 << ' ' << j + 1 << endl;
    char v;
    cin >> v;
    ptr++;
    if (ptr == n)
        ptr = 0;
    lastAns = (v == '<');
    return lastAns;
}

int wait(int x) {
    for (int i = 0; i < x; ++i)
        ask(0, 0);
}

//void waitFor(int ptrG){
//    while (ptr1)
//
//}
//

void solve() {
    cin >> n;
//    for (int i = 0; i < n; ++i)
//        for (int j = i + 1; j < n; ++j) {
//            if (ask(i, j)) {
//                wait()
//            }
//        }
    while (true) {
        bool cng = false, prev = false;

        for (int i = 0; i < n; ++i) {
            prev = !(ask(0, prev ? n - 1 : 0));
            if (prev && i != n - 1)
                cng = true;
        }
        if (cng == false)
            break;
    }
    cout << "!" << endl;
}

int main() {

    ios_base::sync_with_stdio(0);
    cout.tie(0);
    solve();
    return 0;
}
```

Task E

```
// rotating tree

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef pair<ll, ll> pll;

const int MAXN = 200005;


vector<int> g[MAXN];
int sz[MAXN];
int dir[MAXN];

int n, q;

int usedCnt = 0;
vector<bool> used;

int szDfs() {
    vector<int> q;
    used.assign(n, false);
    vector<int> par(n);
    par[0] = 0;
    vector<int> ein;

    while (!q.empty()) {
        int v = q.back();
        q.pop_back();
        ein.push_back(v);
        sz[v] = 1;
        for (auto i : g[v])
            if (i != par[v]) {
                par[i] = v;
                q.push_back(i);
            }
    }
    reverse(ein.begin(), ein.end());
    for (auto i : ein)
        if (i != 0)
            sz[par[i]] += sz[i];
}

vector<int> stP, cycP;


int rightDfs(vector<int> &path) {
    int v = 0;
    path.clear();
```

```
    while (path.size() < (1e6 + 10)) {
        if (!used[v]) {
            usedCnt++;
            used[v] = true;
        }
        if (v == 0 && usedCnt == n)
            return path.size();
        int nxt = g[v][dir[v]];
        dir[v] = (dir[v] + 1) % ((int) g[v].size());
        path.push_back(v);
        v = nxt;
    }
}


mt19937 rnd(std::chrono::system_clock::now().time_since_epoch().count());

void solve() {
    cin >> n >> q;
    fill(dir, dir + MAXN, 0);
    used.resize(n);

    for (int i = 0; i < n; ++i) {
        int cnt = 0;
        cin >> cnt;
        g[i].resize(cnt);
        for (int j = 0; j < cnt; ++j) {
            cin >> g[i][j];
            g[i][j]--;
        }
    }
//    if (rnd() % 6 == 0)
//        return;

    szDfs();

    int stL = rightDfs(stP);
    used.assign(n, 0);
    usedCnt = 0;
    ll cycL = rightDfs(cycP);


    for (int i = 0; i < q; ++i) {
        ll t;
        cin >> t;
        if (t < stL)
            cout << stP[t] + 1 << ' ';
        else {
            t -= stL;
            t %= cycL;
            cout << cycP[t] + 1 << ' ';
        }
    }
```

```
    cout << '\n';
}

int main() {

    ios_base::sync_with_stdio(0);
    cout.tie(0);
    solve();
    return 0;
}
```