Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0130 – Девятьяров Иван Андреевич

A	B	C	D	E	F	sum
30	100	100	75	100	.	405

Task A

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

//#pragma GCC optimize("O3,unroll-loops") // let the chaos begin!
//#pragma GCC target("avx,avx2,bmi,bmi2,popcnt,lzcnt,tune=native")
//#pragma GCC comment(linker, "/stack:20000000000")
//retry

using namespace std;

using ll = long long;
using ld = long double;
using str = string;
using uint128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdb = pair<ld, ld>;

using vin = vector<int>;
using vbo = vector<bool>;
using vll = vector<ll>;
using vdb = vector<ld>;
using vst = vector<string>;
using vpi = vector<pii>;
using vpl = vector<pll>;
using vpd = vector<pdb>;

using vvl = vector<vll>;
```

```
using vvi = vector<vin>;
using vvs = vector<vst>;
using vvd = vector<vdb>;

#define all(x) (x).begin(), (x).end()
#define fi first
#define se second

#define sz(x) int((x).size())
#define sor(x) sort(all(x))
#define rsz resize
#define ins insert
#define ft front()
#define bk back()
#define pub push_back
#define pob pop_back
#define puf push_front
#define pof pop_front
#define rtn return

#define lb lower_bound
#define ub upper_bound

#define FOR(i, a, b) for (ll i = (a); i < (b); ++i)
#define FOR(i, a) FOR(i, 0, a)
#define ROF(i, a, b) for (ll i = (b)-1; i >= (a); --i)
#define ROF(i, a) ROF(i, 0, a)
#define rep(a) FOR(_, a)
#define each(a, x) for (auto &a : x)

//#define max(a, b) a > b ? a : b
//#define min(a, b) a < b ? a : b

#ifdef PERVEEM_LOCAL
#define debug(x) cerr << #x << ": " << x << '\n';
#else
#define debug(x);
#endif

ll C(ll n, ll k) {
    if (k > n){
        return 0;
    }
    if (k == 0){
        return 1;
    }
    return C(n - 1, k - 1) + C(n - 1, k);
}

ll solve1(ll a, ll b, ll n){
    ll ans = 0;
    if (b > n){
        if (n % a == 0){
```

```
            return a;
        }
        return 0;
    }
    FOR(i, n + 1) {
        FOR(j, n + 1) {
            if (i * a + j * b == n) {
                if (i != 0) {
                    ans += C(i + j - 1, min(i - 1, j)) * a;
                }
                if (j != 0) {
                    ans += C(i + j - 1, min(i, j - 1)) * b;
                }
            }
        }
    }
    return ans;
}

ll solve2(ll a, ll b, ll n){
    ll ans = 0;
    FOR(i, 1, (1 << n)) {
        bool is_good = true;
        FOR(j, n) {
            if ((i >> j) & 1) {
                ll cnt = 0;
                FOR(k, n) {
                    cnt++;
                    if ((i >> ((j + k + 1) % n)) & 1) {
                        if (cnt != a && cnt != b) {
                            is_good = false;
                            break;
                        }
                        cnt = 0;
                    }
                }
                break;
            }
        }
        ans += is_good;
    }
    return ans;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    ll a, b, n;
    cin >> a >> b >> n;
    cout << solve1(a, b, n);
}
```

Task B

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

//#pragma GCC optimize("O3,unroll-loops") // let the chaos begin!
//#pragma GCC target("avx,avx2,bmi,bmi2,popcnt,lzcnt,tune=native")
//#pragma GCC comment(linker, "/stack:20000000000")
//retry

using namespace std;

using ll = long long;
using ld = long double;
using str = string;
using uint128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdb = pair<ld, ld>;

using vin = vector<int>;
using vbo = vector<bool>;
using vll = vector<ll>;
using vdb = vector<ld>;
using vst = vector<string>;
using vpi = vector<pii>;
using vpl = vector<pll>;
using vpd = vector<pdb>;

using vvl = vector<vll>;
using vvi = vector<vin>;
using vvs = vector<vst>;
using vvd = vector<vdb>;

#define all(x) (x).begin(), (x).end()
#define fi first
#define se second

#define sz(x) int((x).size())
#define sor(x) sort(all(x))
#define rsz resize
#define ins insert
#define ft front()
#define bk back()
#define pub push_back
#define pob pop_back
#define puf push_front
#define pof pop_front
#define rtn return

#define lb lower_bound
#define ub upper_bound
```

```
#define FOR(i, a, b) for (ll i = (a); i < (b); ++i)
#define FOR(i, a) FOR(i, 0, a)
#define ROF(i, a, b) for (ll i = (b)-1; i >= (a); --i)
#define ROF(i, a) ROF(i, 0, a)
#define rep(a) FOR(_, a)
#define each(a, x) for (auto &a : x)

//#define max(a, b) a > b ? a : b
//#define min(a, b) a < b ? a : b

#ifdef PERVEEM_LOCAL
#define debug(x) cerr << #x << ": " << x << '\n';
#else
#define debug(x);
#endif

//24fin0130
//eversyter

ll n;

ll dist(ll x1, ll y1, ll z1, ll x2, ll y2, ll z2){
    ll d = (abs(x2 - x1) + abs(y2 - y1) + abs(z2 - z1));
    return ((abs(x2 - x1) + abs(y2 - y1) + abs(z2 - z1)) / 2);
}

ll iterate_dist(ll x1, ll y1, ll z1, ll x2, ll y2, ll z2){
    ll ans = 1e16;
    ans = min(ans, dist(x1, y1, z1, x2, y2, z2));
    ans = min(ans, dist(x1, y1, z1, x2 + 2 * n, y2 - n, z2 - n));
    ans = min(ans, dist(x1, y1, z1, x2 - 2 * n, y2 + n, z2 + n));
    ans = min(ans, dist(x1, y1, z1, x2 - n, y2 + 2 * n, z2 - n));
    ans = min(ans, dist(x1, y1, z1, x2 + n, y2 - 2 * n, z2 + n));
    ans = min(ans, dist(x1, y1, z1, x2 - n, y2 - n, z2 + 2 * n));
    ans = min(ans, dist(x1, y1, z1, x2 + n, y2 + n, z2 - 2 * n));
    return ans;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cin >> n;
    ll x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    ll x2, y2, z2;
    cin >> x2 >> y2 >> z2;
    ll ans = 1e16;
    ans = min(ans, iterate_dist(x1, y1, z1, x2, y2, z2));
    ans = min(ans, iterate_dist(x2, y2, z2, x1, y1, z1));
    cout << ans;
}
```

Task C

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

//#pragma GCC optimize("O3,unroll-loops") // let the chaos begin!
//#pragma GCC target("avx,avx2,bmi,bmi2,popcnt,lzcnt,tune=native")
//#pragma GCC comment(linker, "/stack:20000000000")
//retry

using namespace std;

using ll = long long;
using ld = long double;
using str = string;
using uint128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdb = pair<ld, ld>;

using vin = vector<int>;
using vbo = vector<bool>;
using vll = vector<ll>;
using vdb = vector<ld>;
using vst = vector<string>;
using vpi = vector<pii>;
using vpl = vector<pll>;
using vpd = vector<pdb>;

using vvl = vector<vll>;
using vvi = vector<vin>;
using vvs = vector<vst>;
using vvd = vector<vdb>;

#define all(x) (x).begin(), (x).end()
#define fi first
#define se second

#define sz(x) int((x).size())
#define sor(x) sort(all(x))
#define rsz resize
#define ins insert
#define ft front()
#define bk back()
#define pub push_back
#define pob pop_back
#define puf push_front
#define pof pop_front
#define rtn return

#define lb lower_bound
#define ub upper_bound
```

```
#define FOR(i, a, b) for (ll i = (a); i < (b); ++i)
#define FOR(i, a) FOR(i, 0, a)
#define ROF(i, a, b) for (ll i = (b)-1; i >= (a); --i)
#define ROF(i, a) ROF(i, 0, a)
#define rep(a) FOR(_, a)
#define each(a, x) for (auto &a : x)

//#define max(a, b) a > b ? a : b
//#define min(a, b) a < b ? a : b

#ifdef PERVEEM_LOCAL
#define debug(x) cerr << #x << ": " << x << '\n';
#else
#define debug(x);
#endif

ll MOD = 1000000007;

struct mint {
    ll x;

    mint() { x = 0; }

    mint(ll _x) { x = _x; }

    mint operator+(mint other) {
        mint res = *this;
        res.x += other.x;
        if (res.x >= MOD) {
            res.x -= MOD;
        }
        return res;
    }

    mint operator-(mint other) {
        mint res = *this;
        res.x -= other.x;
        res.x += MOD;
        if (res.x >= MOD) {
            res.x -= MOD;
        }
        return res;
    }

    mint operator*(mint other) {
        mint res = *this;
        res.x *= other.x;
        res.x %= MOD;
        return res;
    }

    mint operator<(mint other) {
        return x < other.x;
```

```
    }
};

mint p = mint(3);
vector<mint> vp;

void gen(ll n) {
    vp.resize(n + 1);
    vp[0] = mint(1);
    FOR(i, 1, n + 1) {
        vp[i] = vp[i - 1] * p;
    }
}

void solve2() {
    ll n;
    cin >> n;
    string s;
    cin >> s;
    ll act_mn, act_mx;
    cin >> act_mn >> act_mx;
    vector<mint> ps(n + 1);
    ps[0] = mint(0);
    FOR(i, n) {
        ps[i + 1] = ps[i] * p + mint(s[i] - '0' + 1);
    }
    mint mn = ps[n];
    mint mx = ps[n];
    ROF(i, 1, n) {
        mint h = ps[i] + (ps[n] - ps[i] * vp[n - i]) * vp[i];
        if (mn.x > h.x) {
            mn = h;
        }
        if (mx.x < h.x) {
            mx = h;
        }
    }
    if (mn.x == act_mn && mx.x == act_mx) {
        cout << "YES";
    } else {
        cout << "NO";
    }
}

void solve1() {
    ll n;
    cin >> n;
    string s;
    cin >> s;
    vector<mint> ps(n + 1);
    ps[0] = mint(0);
    FOR(i, n) {
        ps[i + 1] = ps[i] * p + mint(s[i] - '0' + 1);
```

```
    }
    mint mn = ps[n];
    mint mx = ps[n];
    ROF(i, 1, n) {
        mint h = ps[i] + (ps[n] - ps[i] * vp[n - i]) * vp[i];
        if (mn.x > h.x) {
            mn = h;
        }
        if (mx.x < h.x) {
            mx = h;
        }
    }
    cout << mn.x << ' ' << mx.x;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    gen(1000000);
    string t;
    cin >> t;
    if (t == "first") {
        solve1();
    } else {
        solve2();
    }
}
```

Task D

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

//#pragma GCC optimize("O3,unroll-loops") // let the chaos begin!
//#pragma GCC target("avx,avx2,bmi,bmi2,popcnt,lzcnt,tune=native")
//#pragma GCC comment(linker, "/stack:20000000000")
//retry

using namespace std;

using ll = long long;
using ld = long double;
using str = string;
using uint128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdb = pair<ld, ld>;

using vin = vector<int>;
using vbo = vector<bool>;
using vll = vector<ll>;
using vdb = vector<ld>;
```

```
using vst = vector<string>;
using vpi = vector<pii>;
using vpl = vector<pll>;
using vpd = vector<pdb>;

using vvl = vector<vll>;
using vvi = vector<vin>;
using vvs = vector<vst>;
using vvd = vector<vdb>;

#define all(x) (x).begin(), (x).end()
#define fi first
#define se second

#define sz(x) int((x).size())
#define sor(x) sort(all(x))
#define rsz resize
#define ins insert
#define ft front()
#define bk back()
#define pub push_back
#define pob pop_back
#define puf push_front
#define pof pop_front
#define rtn return

#define lb lower_bound
#define ub upper_bound

#define FOR(i, a, b) for (ll i = (a); i < (b); ++i)
#define FOR(i, a) FOR(i, 0, a)
#define ROF(i, a, b) for (ll i = (b)-1; i >= (a); --i)
#define ROF(i, a) ROF(i, 0, a)
#define rep(a) FOR(_, a)
#define each(a, x) for (auto &a : x)

//#define max(a, b) a > b ? a : b
//#define min(a, b) a < b ? a : b

#ifdef PERVEEM_LOCAL
#define debug(x) cerr << #x << ": " << x << '\n';
#else
#define debug(x);
#endif

vll cur = {3, 1, 2, 4};

bool ask_test(ll i, ll j) {
    vll start = cur;
    swap(cur[i], cur[j]);
    ll p = cur[0];
    FOR(k, sz(cur)) {
        cur[k] = cur[k + 1];
```

```
    }
    cur[sz(cur) - 1] = p;
    return cur > start;
}

bool ask_it(ll i, ll j) {
    cout << i + 1 << ' ' << j + 1 << endl;
    char x;
    cin >> x;
    return x == '>';
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    ll n;
    cin >> n;
    vll arr(n);
    FOR(i, n) {
        ll cnt = n - 1;
        FOR(j, n - 1) {
            cnt -= ask_it(0, 0);
            ask_it(0, 0);
            ask_it(n - 2, n - 1);
            FOR(temp, n - 2) {
                ask_it(0, 0);
            }
        }
        ask_it(0, 0);
        arr[i] = cnt;
    }
    vll id(n);
    FOR(i, n) {
        id[arr[i]] = i;
    }
    FOR(i, n) {
        ask_it(id[n - i - 1], n - 1);
        swap(arr[id[n - i - 1]], arr[n - 1]);
        swap(id[n - i - 1], id[arr[id[n - i - 1]]]);
        ll p = arr[0];
        FOR(j, n - 1) {
            arr[j] = arr[j + 1];
        }
        arr[n - 1] = p;
        FOR(j, n) {
            id[j]--;
            id[j] += n;
            id[j] %= n;
        }
    }
    FOR(i, n - 1) {
        ask_it(0, 0);
```

```
    }
    //cout << "";
}
```

Task E

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

#pragma GCC optimize("O3,unroll-loops") // let the chaos begin!
//#pragma GCC target("avx,avx2,bmi,bmi2,popcnt,lzcnt,tune=native")
//#pragma GCC comment(linker, "/stack:20000000000")
//retry

using namespace std;

using ll = long long;
using ld = long double;
using str = string;
using uint128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdb = pair<ld, ld>;

using vin = vector<int>;
using vbo = vector<bool>;
using vll = vector<ll>;
using vdb = vector<ld>;
using vst = vector<string>;
using vpi = vector<pii>;
using vpl = vector<pll>;
using vpd = vector<pdb>;

using vvl = vector<vll>;
using vvi = vector<vin>;
using vvs = vector<vst>;
using vvd = vector<vdb>;

#define all(x) (x).begin(), (x).end()
#define fi first
#define se second

#define sz(x) int((x).size())
#define sor(x) sort(all(x))
#define rsz resize
#define ins insert
#define ft front()
#define bk back()
#define pub push_back
#define pob pop_back
#define puf push_front
#define pof pop_front
```

```
#define rtn return

#define lb lower_bound
#define ub upper_bound

#define FOR(i, a, b) for (ll i = (a); i < (b); ++i)
#define FOR(i, a) FOR(i, 0, a)
#define ROF(i, a, b) for (ll i = (b)-1; i >= (a); --i)
#define ROF(i, a) ROF(i, 0, a)
#define rep(a) FOR(_, a)
#define each(a, x) for (auto &a : x)

//#define max(a, b) a > b ? a : b
//#define min(a, b) a < b ? a : b

#ifdef PERVEEM_LOCAL
#define debug(x) cerr << #x << ": " << x << '\n';
#else
#define debug(x);
#endif

mt19937 rnd(3876571);

struct ordered_set {
    struct Node {
        Node *l = nullptr, *r = nullptr;
        ll x, y, v, cnt;

        Node() {}

        Node(ll _x, ll _v) { x = _x, y = rnd(), v = _v, cnt = 1; }
    };

    ll cnt(Node *v) {
        return v ? v->cnt : 0;
    }

    void update(Node *v) {
        if (v) {
            v->cnt = 1 + cnt(v->l) + cnt(v->r);
        }
    }

    Node *merge(Node *l, Node *r) {
        if (!l) return r;
        if (!r) return l;
        if (l->y > r->y) {
            l->r = merge(l->r, r);
            update(l);
            return l;
        } else {
            r->l = merge(l, r->l);
            update(r);
```

```
            return r;
        }
    }

    using pNN = pair<Node *, Node *>;

    pNN split(Node *v, ll x) {
        if (!v) return {nullptr, nullptr};
        if (v->x > x) {
            pNN d = split(v->l, x);
            v->l = d.second;
            update(v);
            return {d.first, v};
        } else {
            pNN d = split(v->r, x);
            v->r = d.first;
            update(v);
            return {v, d.second};
        }
    }

    Node *root = nullptr;

    void insert(ll x, ll v) {
        Node *d = new Node(x, v);
        pNN q = split(root, x);
        root = merge(q.first, merge(d, q.second));
    }

    ll size() {
        return cnt(root);
    }

    ll get(ll ind, Node *v = nullptr) {
        if (v == nullptr) {
            v = root;
        }
        if (cnt(v->l) > ind) {
            return get(ind, v->l);
        }
        ind -= cnt(v->l);
        if (ind == 0) {
            return v->v;
        }
        ind--;
        return get(ind, v->r);
    }

    ordered_set() {}
};

vector<vll> tree;
vll point;
```

```
vll visit;
vector<vbo> used;
vll parent, parent_id;
vector<vll> tin;
ll timer = 0;

void dfs_parent(ll v, ll p = -1) {
    parent[v] = p;
    each(u, tree[v]) {
        if (u == p) {
            continue;
        }
        dfs_parent(u, v);
    }
}

void calc_tin(ll v) {
    tin[v].pub(timer++);
    used[v].pub(false);
    ll ind = 0;
    FOR(j, sz(tree[v])) {
        if (tree[v][j] == parent[v]) {
            parent_id[v] = j;
            ind = j + 1;
            break;
        }
    }
    ind %= sz(tree[v]);
    FOR(i, sz(tree[v]) - 1) {
        calc_tin(tree[v][ind]);
        ind++;
        ind %= sz(tree[v]);
        tin[v].pub(timer++);
        used[v].pub(false);
    }
}

ordered_set st;
vll new_next, cur_next;

void dfs(ll v, bool got_token = false) {
    if (!used[v][0]) {
        st.insert(tin[v][0], v);
        if (v != 0 && !got_token) {
            new_next.pub(v);
            got_token = true;
        }
    }
    used[v][0] = true;
    while (tree[v][point[v]] != parent[v]) {
        dfs(tree[v][point[v]], got_token);
        if (!used[v][(point[v] - parent_id[v] + sz(tree[v])) % sz(tree[v
    ])]) {
```

```
            used[v][(point[v] - parent_id[v] + sz(tree[v])) % sz(tree[v])
    ] = true;
            st.insert(tin[v][(point[v] - parent_id[v] + sz(tree[v])) % sz
    (tree[v])], v);
        }
        point[v]++;
        point[v] %= sz(tree[v]);
    }
    point[v]++;
    point[v] %= sz(tree[v]);
}

void check_dfs(ll v, ll p = -1) {
    timer--;
    if (timer == 0) {
        cout << v + 1 << ' ';
        return;
    }
    while (tree[v][point[v]] != p) {
        check_dfs(tree[v][point[v]], v);
        if (timer == 0) {
            return;
        }
        timer--;
        if (timer == 0) {
            cout << v + 1 << ' ';
            return;
        }
        point[v]++;
        point[v] %= sz(tree[v]);
        if (tree[v][point[v]] == -1) {
            point[v] = 0;
        }
    }
    point[v]++;
    point[v] %= sz(tree[v]);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    ll n, q;
    cin >> n >> q;
    point.resize(n, 0);
    tree.resize(n);
    used.resize(n);
    parent_id.resize(n);
    parent.resize(n);
    tin.resize(n);
    FOR(i, n) {
        ll k;
        cin >> k;
```

```
        FOR(j, k) {
            ll x;
            cin >> x;
            x--;
            tree[i].pub(x);
        }
    }
    tree[0].pub(-1);
    dfs_parent(0);
    calc_tin(0);
    used[0].back() = true;
    vpl requests;
    FOR(i, q) {
        ll t;
        cin >> t;
        requests.pub({t, i});
    }
    vll ans(q, -1);
    sor(requests);
    ll add = 0;
    new_next.pub(0);
    FOR(i, q) {
        while (add + st.size() <= requests[i].first && sz(new_next) > 0)
   {
            add += st.size();
            cur_next = new_next;
            new_next.clear();
            each(v, cur_next) {
                dfs(v);
            }
        }
        requests[i].first -= add;
        requests[i].first %= st.size();
        cout << st.get(requests[i].first) + 1 << ' ';
    }
}
```