Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0254 – Курлаев Никита Витальевич

A	B	C	D	E	F	sum
100	50	100	100	20	47	417

Task A

```
// #pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

const ll maxn = 100;

ll dp[maxn][maxn];

void solve() {
    ll a, b, n;
    cin >> a >> b >> n;

    if (a > b) swap(a, b);

    dp[0][0] = 1;

    for (ll i = 0; i <= n; i++) {
```

```
        for (ll j = 1; j <= n; j++) {
            if (i - a >= 0) dp[i][j] += dp[i-a][j-1];
            if (a != b && i - b >= 0) {
                dp[i][j] += dp[i-b][j-1];
            }
        }
    }

    ll ans = 0;
    for (ll i = 1; i <= n; i++) {
        ans += (dp[n][i] * n) / i;
    }

    cout << ans << endl;
}

int main() {
    cin.tie(nullptr);
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```

Task B

```
// #pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

struct pt3 {
    ll x, y, z;

    ll dist(const pt3 &o) const {
        ll d = abs(x - o.x) + abs(y - o.y) + abs(z - o.z);
        return d / 2;
    }

    static pt3 read() {
        pt3 p;
        cin >> p.x >> p.y >> p.z;
```

```
        return p;
    }
};

void solve() {
    ll n;
    cin >> n;

    pt3 p1 = pt3::read();
    pt3 p2 = pt3::read();

    cout << p1.dist(p2) << endl;
}

int main() {
    cin.tie(nullptr);
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```

Task C

```
// #pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

ll mod(ll x, ll m) {
    return (x % m + m) % m;
}

struct hasher {
    ll n, P, M;
    vector<ll> p, h;

    hasher(ll P, ll M, string& s): n(s.size()), P(P), M(M) {
        p.assign(n + 1, 1);
        p[0] = 1;
        for (ll i = 1; i <= n; i++) {
            p[i] = mod(p[i-1] * P, M);
        }
```

```
        h.assign(n + 1, 0);
        h[0] = mod(p[n-1] * s[0], M);
        for (ll i = 1; i < n; i++) {
            h[i] = mod(h[i-1] + mod(p[n-i-1] * s[i], M), M);
        }
    }

    ll hash_sub(ll i, ll sz) {
        if (i == 0) {
            return h[sz - 1];
        } else {
            return mod(mod(h[i + sz - 1] - h[i - 1], M) * p[i], M);
        }
    }

    ll hash_all() {
        return h.back();
    }
};

// 6 79461 79461 79461 79461

vector<ll> mds = {
    ll(1e9) + 7,
    ll(1e9) + 9,
    ll(1e9) + 21,
    ll(1e9) + 33,
    // ll(1e9) + 7
};

ll p0 = 1009;

// 6 79461 79461 79461 79461
// 6 433675845 321881664 651135331 980421134

void first() {
    ll n;
    cin >> n;
    string s;
    cin >> s;

    s += s;

    // for (ll i = 0; i < s.size(); i++) {
    //     s[i] -= '0';
    // }

    cout << n << ' ';
    for (ll m: mds) {
        hasher h0(p0, m, s);
        cout << h0.hash_sub(0, n) << ' ';
    }
```

```
    cout << endl;
}

void second() {
    ll n;
    cin >> n;
    string t;
    cin >> t;

    ll old_n;
    cin >> old_n;

    vector<ll> hss(mds.size());
    for (ll &i: hss) cin >> i;

    if (n != old_n) {
        cout << "NO" << endl;
        return;
    }

    // cout << "YES" << endl; return;

    t += t;

    // for (auto &i: t) i -= '0';

    vector<hasher> z;
    for (ll m: mds) {
        z.push_back(hasher(p0, m, t));
    }

    for (ll i = 0; i < n; i++) {
        bool ok = true;
        for (ll j = 0; j < mds.size(); j++) {
            ok &= (z[j].hash_sub(i, n) == hss[j]);
        }
        if (ok) {
            cout << "YES" << endl;
            return;
        }
    }

    cout << "NO" << endl;
    return;
}

void solve() {
    string l;
    cin >> l;

    if (l == "first") {
        first();
    } else {
```

```
        second();
    }
}

int main() {
    cin.tie(nullptr);
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```

Task D

```
#pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

vector<ll> p;
ll n;

bool act(ll a, ll b) {
    cout << a + 1 << ' ' << b + 1 << endl;
    swap(p[a], p[b]);
    rotate(p.begin(), p.begin() + 1, p.end());

    // cerr << "now ";
    // for (ll &i: p) cerr << i << ' ';
    // cerr << endl;

    string s;
    cin >> s;

    if (s == "!") exit(0);

    return s == "<";
}

ll pos(ll x) {
    return find(p.begin(), p.end(), x) - p.begin();
}
```

```
bool cmp2(ll a, ll b) {
    while (p[0] != a) {
        act(0, 0);
    }

    bool res = !act(1, pos(b)); // mega sus !
    // cerr << "- " << a << (res ? " < " : " > ") << b << endl;
    return res;
}

bool cmp3(ll a, ll b) {
    if (p[0] != a) act(1, pos(a));

    return !act(1, pos(b));
}

bool cmp(ll a, ll b) {
    while (p[0] != a && p[0] != b) {
        act(0, 0);
    }

    if (p[0] == a) {
        return !act(1, pos(b)); // mega sus !
    } else {
        return act(1, pos(a));
    }
}

void solve() {
    cin >> n;

    p.resize(n);

    iota(p.begin(), p.end(), 0ll);

    auto p2 = p;

    sort(p2.begin(), p2.end(), [](ll a, ll b) {
        return cmp3(a, b);
    });

    vector<ll> p_orig(n, 0);
    for (ll i = 0; i < n; i++) {
        p_orig[p2[i]] = i;
    }

    // cerr << "FOUND" << endl;
    // for (ll &i: p_orig) cerr << i << ' ';
    // cerr << endl;
    // for (ll &i: p) cerr << i << ' ';
    // cerr << endl << endl;
```

```
    vector<ll> p_tgt = p2;
    vector<ll> from = p;
    reverse(p_tgt.begin(), p_tgt.end());

    // ll bgn = pos(p_tgt[0]);
    for (ll i = 1; i < n; i++) {
        ll b0 = pos(p_tgt[0]);
        ll need = (b0 + i) % n;
        ll have = pos(p_tgt[i]);
        // for (ll &i: p) cerr << i << ' ';
        // cerr << " => " << need << " " << have << endl;
        if (need != have) {
            act(have, need);
        }
    }

    while (pos(p_tgt[0]) != 0) act(0, 0);

    // cout << "!" << endl;
}

int main() {
    cin.tie(nullptr);
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```

Task E

```
// #pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

const ll maxn = 1e5 + 1;

vector<ll> adj[maxn];
vector<ll> pos;

vector<ll> ans;
```

```
// ll tr = 0;

void solve() {
    ll n, q;
    cin >> n >> q;

    pos.resize(n);

    for (ll i = 0; i < n; i++) {
        ll m;
        cin >> m;

        adj[i].resize(m);
        for (ll &j: adj[i]) {
            cin >> j;
            j--;
        }
    }

    // vector<ll> qs(q);

    // for (ll &i: qs) cin >> i;

    ll v = 0;
    while (ans.size() < 1e6) {
        ans.push_back(v);

        ll nxt = adj[v][pos[v]];
        pos[v] = (pos[v] + 1) % ll(adj[v].size());
        v = nxt;
    }

    for (ll i = 0; i < q; i++) {
        ll t;
        cin >> t;

        cout << ans[t] + 1 << ' ';
    }
    cout << endl;
}

int main() {
    cin.tie(nullptr);
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```

Task F

```
// #pragma GCC optimize("Ofast")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <iostream>
```

```cpp
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <numeric>
#include <algorithm>
#include <random>

using namespace std;
using ll = long long;
using ull = unsigned long long;
using fl = long double;

void solve() {
    ll h, w;
    cin >> h >> w;

    vector<string> a;

    for (ll i = 0; i < h; i++) {
        string s;
        cin >> s;

        a.push_back(s);
    }

    ll ans = 0;
    for (ll i = 0; i < h; i++) {
        for (ll j = 0; j < w; j++) {
            if (j % 2 == 0 && j < w - 1) {
                ans += ll(a[i][j] - '0') * ll(a[i][j+1] - '0');
            }
        }
    }

    cout << ans << endl;

    for (ll i = 0; i < h; i++) {
        for (ll j = 0; j < w; j++) {
            if (j % 2 == 0 && j < w - 1) {
                cout << 'L';
            } else {
                cout << 'R';
            }
        }

        cout << '\n';
    }
    cout << flush;
}

int main() {
    cin.tie(nullptr);
```

```
    ios::sync_with_stdio(false);
    // ll t; cin >> t; while (t --> 0)
    solve();
}
```