Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0214 – Козлов Виталий Сергеевич

A	B	C	D	E	F	sum
100	100	100	100	40	93	533

Task A

```
cache = {}


def solve(x):
    if x == 0:
        return 1
    if x < 0:
        return 0
    if x in cache:
        return cache[x]
    result = 0
    if x >= a:
        result += solve(x - a)
    if x >= b:
        result += solve(x - b)
    cache[x] = result
    return result


a, b, n = map(int, input().split())
print(solve(n - a) * a + solve(n - b) * b)
```

Task B

```
directions = [(1, 0, -1), (0, 1, -1), (-1, 1, 0), (-1, 0, 1), (0, -1, 1),
     (1, -1, 0)]


def get_reversed(x1, y1, z1):
```

```
    if (abs(x1), abs(y1), abs(z1)).count(n) == 2:
        result = []
        for i in range(3):
            for j in range(i + 1, 3):
                result.append([-x1, -y1, -z1])
                result[-1][i], result[-1][j] = result[-1][j], result[-1][
    i]
                result[-1] = tuple(result[-1])
        return result

    result = []
    if abs(x1) == n:
        result.append((-x1, -z1, -y1))
    if abs(y1) == n:
        result.append((-z1, -y1, -x1))
    if abs(z1) == n:
        result.append((-y1, -x1, -z1))
    return result


def dist(x1, y1, z1, x2, y2, z2):
    x2 -= x1
    y2 -= y1
    z2 -= z1
    result = abs(x2) + (abs(y2) + abs(z2) - abs(x2) + 1) // 2
    return result


def try_dist(x1, y1, z1, x2, y2, z2):
    result = dist(x1, y1, z1, x2, y2, z2)
    for a, b, c in directions:
        dst = min(
            10 ** 9 if a == 0 else n - x1 if a == 1 else abs(-n - x1),
            10 ** 9 if b == 0 else n - y1 if b == 1 else abs(-n - y1),
            10 ** 9 if c == 0 else n - z1 if c == 1 else abs(-n - z1),
        )
        nx, ny, nz = x1 + a * dst, y1 + b * dst, z1 + c * dst
        for V in get_reversed(nx, ny, nz):
            result = min(result, dist(x1, y1, z1, nx, ny, nz) + dist(*V,
    x2, y2, z2))

        dst = min(
            10 ** 9 if a == 0 else n - x2 if a == 1 else abs(-n - x2),
            10 ** 9 if b == 0 else n - y2 if b == 1 else abs(-n - y2),
            10 ** 9 if c == 0 else n - z2 if c == 1 else abs(-n - z2),
        )
        nx, ny, nz = x2 + a * dst, y2 + b * dst, z2 + c * dst
        for V in get_reversed(nx, ny, nz):
            result = min(result, dist(x2, y2, z2, nx, ny, nz) + dist(*V,
    x1, y1, z1))
    return result
```

```
def bfs(x1, y1, z1, x2, y2, z2):
    dst = {(x1, y1, z1): 0}
    q = [(x1, y1, z1)]
    if n in (abs(x1), abs(y1), abs(z1)):
        for V in get_reversed(x1, y1, z1):
            dst[V] = 0
            q.append(V)
    for i, j, k in q:
        for a, b, c in directions:
            if abs(i + a) <= n and abs(j + b) <= n and abs(k + c) <= n
    and (
                    (i + a, j + b, k + c) not in dst or dst[(i, j, k)] +
    1 < dst[(i + a, j + b, k + c)]):
                dst[(i + a, j + b, k + c)] = dst[(i, j, k)] + 1
                q.append((i + a, j + b, k + c))
                if n in (abs(i + a), abs(j + b), abs(k + c)):
                    for V in get_reversed(i + a, j + b, k + c):
                        if V not in dst or dst[(i + a, j + b, k + c)] <
    dst[V]:
                            dst[V] = dst[(i + a, j + b, k + c)]
                            q.append(V)
    return dst[(x2, y2, z2)]


n = int(input())
x1, y1, z1 = map(int, input().split())
x2, y2, z2 = map(int, input().split())
if n <= 10:
    print(bfs(x1, y1, z1, x2, y2, z2))
    exit(0)

print(try_dist(x1, y1, z1, x2, y2, z2))
```

Task C

```
#include <bits/stdc++.h>
#ifdef LOCAL
#include "printer.hpp"
#else
# define print(...)
#endif
using namespace std;
bool __nothing0 = ios::sync_with_stdio(false);
bool __nothing1 = cin.tie(0);
bool __nothing2 = cout.tie(0);
# define int long long
const int MAXN = 1e5 + 1, MOD = 1e9 + 7, p = 11, p2 = 359;
pair<int, unsigned int> power[MAXN];
int power2[MAXN];

struct HashString {
    vector<pair<int, unsigned int>> h;
    vector<pair<int, int>> h2;
```

```
    HashString(const string &s) {
        if (power[0].first == 0 && power[0].second == 0) {
            power[0] = {1, 1};
            for (int i = 1; i < MAXN; i++) {
                power[i].first = power[i - 1].first * p % MOD;
                power[i].second = power[i - 1].second * p;
            }
            power2[0] = 1;
            for (int i = 1; i < MAXN; i++) {
                power2[i] = power2[i - 1] * p2 % MOD;
            }
        }

        h.resize(s.size() + 1);
        h[0] = {0, 0};
        for (int i = 0; i < (int) s.size(); i++) {
            h[i + 1].first = (h[i].first * p + (s[i] - '0' + 1)) % MOD;
            h[i + 1].second = h[i].second * p + (s[i] - '0' + 1);
        }
        h2.resize(s.size() + 1);
        h2[0] = {0, 0};
        for (int i = 0; i < (int) s.size(); i++) {
            h2[i + 1].first = (h2[i].first * p2 + (s[i] - '0' + 1)) % MOD
    ;
            h2[i + 1].second = (h2[i].second * p2 + (s[i] - '0' + 1)) %
    MOD;
        }
    }

    pair<pair<int, unsigned int>, pair<int, int>> get(int l, int r) {
        return {{
                        (h[r].first - (h[l].first * power[r - l].first) %
     MOD + MOD) % MOD,
                        h[r].second - h[l].second * power[r - l].second
                },
                {
                        (h2[r].first - (h2[l].first * power2[r - l]) %
    MOD + MOD) % MOD,
                        (h2[r].second - (h2[l].second * power2[r - l]) %
    MOD + MOD) % MOD
                }};
    }
};

int32_t main() {
    int n;
    string o, s;
    cin >> o >> n >> s;
    if (o == "first") {
        HashString h(s);
        auto res = h.get(0, n);
        cout << res.first.first << " " << res.first.second << " " << res.
```

```
    second.first << " " << res.second.second
              << endl;
    } else {
        HashString h(s + s);
        pair <pair<int, unsigned int>, pair<int, int>> old_h;
        cin >> old_h.first.first >> old_h.first.second >> old_h.second.
    first >> old_h.second.second;
        for (int i = 0; i <= n; i++) {
            if (h.get(i, i + n) == old_h) {
                cout << "YES" << endl;
                return 0;
            }
        }
        cout << "NO" << endl;
    }
}
```

Task D

```
#include <bits/stdc++.h>
#ifdef LOCAL
#include "printer.hpp"
#else
# define print(...)
#endif
using namespace std;
bool __nothing0 = ios::sync_with_stdio(false);
bool __nothing1 = cin.tie(0);
bool __nothing2 = cout.tie(0);
# define int long long
int pos[501], add = 0, n;

int get_pos(int i) {
    return ((pos[i] - add) % n + n) % n + 1;
}

int find(int x) {
    for (int i = 0; i < n; i++) {
        if (get_pos(i) == x) {
            return i;
        }
    }
}

bool cmp(int a, int b) {
    char res;
    cout << get_pos(b) << " " << 2 << endl;
    swap(pos[b], pos[find(2)]);
    cin >> res;
    if (res == '!') {
        exit(0);
    }
    add++;
```

```
    cout << get_pos(a) << " " << 2 << endl;
    swap(pos[a], pos[find(2)]);
    cin >> res;
    if (res == '!') {
        exit(0);
    }
    add++;
    return res == '<';
}

vector<int> qsort(vector<int> A) {
    if (A.size() <= 1) {
        return A;
    }
    int v = A[A.size() / 2];
    vector<int> left, right, result;
    for (int x : A) {
        if (x == v) continue;
        if (cmp(x, v)) {
            left.push_back(x);
        } else {
            right.push_back(x);
        }
    }
    left = qsort(left);
    right = qsort(right);
    result.insert(result.end(), left.begin(), left.end());
    result.push_back(v);
    result.insert(result.end(), right.begin(), right.end());
    return result;
}

int32_t main() {
    cin >> n;
    vector<int> numbers;
    for (int i = 0; i < n; i++) {
        pos[i] = i;
        numbers.push_back(i);
    }
    numbers = qsort(numbers);
    reverse(numbers.begin(), numbers.end());
    for (int i = 0; i < n; i++) {
        cout << get_pos(numbers[i]) << " " << 1 << endl;
        char res;
        cin >> res;
        if (res == '!') {
            exit(0);
        }
        swap(pos[numbers[i]], pos[find(1)]);
        add++;
    }
}
```

Task E

```
#include <bits/stdc++.h>
#ifdef LOCAL
#include "printer.hpp"
#else
# define print(...)
#endif
using namespace std;
bool __nothing0 = ios::sync_with_stdio(false);
bool __nothing1 = cin.tie(0);
bool __nothing2 = cout.tie(0);
# define int long long
const int MAXN = 1e5 + 1;
vector<int> d[MAXN];


int32_t main() {
    int n, q;
    cin >> n >> q;
    for (int i = 1; i <= n; i++) {
        int sz;
        cin >> sz;
        d[i].resize(sz);
        for (int &x : d[i]) {
            cin >> x;
        }
    }
    if (n <= 1000) {
        vector<int> add(n + 1), ans;
        int cycle_start = -1;
        int v = 1;
        for (int i = 0; i < n * n + 10; i++) {
            ans.push_back(v);
            int u = d[v][(add[v] % d[v].size() + d[v].size()) % d[v].size
    ()];
            add[v]++;
            v = u;
        }
        for (int sz = 2; sz < ans.size() / 3; sz++) {
            bool flag = false;
            for (int i = 0; i <= sz; i++) {
                if (ans[ans.size() - i - 1] != ans[ans.size() - i - 1 -
    sz]) {
                    flag = true;
                    break;
                }
            }
            if (!flag) {
                cycle_start = ans.size() - sz;
                break;
            }
        }
```

```
        while (q--) {
            int t;
            cin >> t;
            if (t < cycle_start) {
                cout << ans[t] << " ";
            } else {
                cout << ans[cycle_start + (t - cycle_start) % (ans.size()
 - cycle_start)] << " ";
            }
        }

        return 0;
    }

    vector<int> add(n + 1), ans(1e6 + 1);
    int v = 1;
    for (int t = 0; t < ans.size(); t++) {
        ans[t] = v;
        int u = d[v][(add[v] % d[v].size() + d[v].size()) % d[v].size()];
        add[v]++;
        v = u;
    }

    while (q--) {
        int t;
        cin >> t;
        cout << ans[t] << " ";
    }
}
```

Task F

```
#pragma GCC optimize("O3")
#include <bits/stdc++.h>
#ifdef LOCAL
#include "printer.hpp"
#else
# define print(...)
#endif
using namespace std;
bool __nothing0 = ios::sync_with_stdio(false);
bool __nothing1 = cin.tie(0);
bool __nothing2 = cout.tie(0);
int h, w;
vector<vector<int>> A;

vector<vector<char>> ans;
int res = -1;
mt19937 rnd(42);

void try_solve() {
    vector<vector<char>> cur_ans(h, vector<char>(w, '.'));
```

```
    int cur_res = 0;
    while (true) {
        map<int, vector<tuple<int, int, char, int, int, char>>> cur;
        for (int i = 0; i < h; i++) {
            for (int j = 0; j < w; j++) {
                if (cur_ans[i][j] == '.' && cur_ans[(i + 1) % h][j] == '.
    ') {
                    cur[A[i][j] * A[(i + 1) % h][j]].emplace_back(i, j, '
    T', (i + 1) % h, j, 'B');
                }
                if (cur_ans[i][j] == '.' && cur_ans[i][(j + 1) % w] == '.
    ') {
                    cur[A[i][j] * A[i][(j + 1) % w]].emplace_back(i, j, '
    L', i, (j + 1) % w, 'R');
                }
            }
        }
        if (cur.empty()) {
            break;
        }
        auto [x1, y1, c1, x2, y2, c2] = cur.rbegin()->second[rnd() % cur.
    rbegin()->second.size()];
        cur_res += cur.rbegin()->first;
        cur_ans[x1][y1] = c1;
        cur_ans[x2][y2] = c2;
    }
    if (cur_res > res) {
        res = cur_res;
        ans = cur_ans;
    }
}

int32_t main() {
    auto start = clock();
    cin >> h >> w;
    A.resize(h, vector<int>(w));
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            char c;
            cin >> c;
            A[i][j] = c - '0';
        }
    }
    while (clock() - start < 0.95 * CLOCKS_PER_SEC) {
        try_solve();
    }
    cout << res << "\n";
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << ans[i][j];
        }
        cout << "\n";
    }
```

```
    return 0;
}
```