Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0507 – Шафранов Артем Андреевич

A	B	C	D	E	F	sum
100	100	100	75	20	49	444

Task A

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

using namespace std;

#define int long long

#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

void solve() {
    int a, b, n;
```

```
    cin >> a >> b >> n;
    vector<int> dp(n + 1, 0);
    dp[0] = 1;
    for (int i = min(a, b); i <= n; i++) {
        if (i - a >= 0)
            dp[i] += dp[i - a];
        if (i - b >= 0)
            dp[i] += dp[i - b];
    }
    int ans = dp[n];
    if (n - a >= 0)
        for (int i = 1; i < a; i++)
            ans += dp[n - a];
    if (a != b && n - b >= 0)
        for (int i = 1; i < b; i++)
            ans += dp[n - b];
    cout << ans;
}

signed main() {
    solve();
}
```

Task B

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

using namespace std;

#define int long long

#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

const int INF = 1e18;
```

```
int get_dist(int x1, int y1, int z1, int x2, int y2, int z2) {
    int dx = x1 - x2, dy = y1 - y2, dz = z1 - z2;
    vector<int> sorted {dx, dy, dz};
    sort(walk(sorted));
    int f = abs(sorted[0]), s = abs(sorted[2]);
    return max(f, s);
}

void solve() {
    int n;
    cin >> n;
    int x1, y1, z1, x2, y2, z2;
    cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
    vector<int> orig{x2, y2, z2};
    int ans = get_dist(x1, y1, z1, x2, y2, z2);
    vector<vector<int>> points;
    points.push_back(vector<int>{x2 + 2 * n, y2 - n, z2 - n});
    points.push_back(vector<int>{x2 - 2 * n, y2 + n, z2 + n});
    points.push_back(vector<int>{x2 + n, y2 - 2 * n, z2 + n});
    points.push_back(vector<int>{x2 - n, y2 + 2 * n, z2 - n});
    points.push_back(vector<int>{x2 + n, y2 + n, z2 - 2 * n});
    points.push_back(vector<int>{x2 - n, y2 - n, z2 + 2 * n});
    for (auto u : points)
        ans = min(ans, get_dist(x1, y1, z1, u[0], u[1], u[2]));
    cout << ans;
}

signed main() {
    solve();
}
```

Task C

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

using namespace std;

#define int long long
```

```
#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

const int INF = 1e18, N = 2e5 + 228;

int k[3]{31, 29, 3}, mod[3] {(int)1e9 + 7, (int)1e9 + 7, (int)1e9 + 9};

int hashes[3][N], powed[3][N];

void print(int x) {
    for (int i = 32; i >= 0; i--)
        cout << ((x >> i) & 1);
}

vector<int> get_substr(int l, int r) {
    vector<int> ans;
    for (int j = 0; j < 3; j++)
        ans.push_back((hashes[j][r] + mod[j] - (hashes[j][l - 1] * powed[
    j][r - l + 1] % mod[j])) % mod[j]);
    return ans;
}

vector<int> cut(const string& s) {
    int idx = 0;
    vector<int> ans(3);
    for (int i = idx; i < idx + 33; i++)
        ans[0] += (s[i] - '0') * (1ll << (32 - (i - idx)));
    idx += 33;
    for (int i = idx; i < idx + 33; i++)
        ans[1] += (s[i] - '0') * (1ll << (32 - (i - idx)));
    idx += 33;
    for (int i = idx; i < idx + 33; i++)
        ans[2] += (s[i] - '0') * (1ll << (32 - (i - idx)));
    return ans;
}

void solve() {
    string cmd;
    int n;
    string s;
    cin >> cmd >> n >> s;
    if (cmd == "first") {
        hashes[0][0] = hashes[1][0] = hashes[2][0] = s[0] - '0' + 1;
        for (int i = 1; i < n; i++)
            for (int j = 0; j < 3; j++)
                hashes[j][i] = (hashes[j][i - 1] * k[j] + s[i] - '0' + 1)
     % mod[j];
        print(hashes[0][n - 1]);
```

```
        print(hashes[1][n - 1]);
        print(hashes[2][n - 1]);
    }
    else {
        string t;
        cin >> t;
        string doubled_s = s + s;
        hashes[0][0] = hashes[1][0] = hashes[2][0] = doubled_s[0] - '0' +
 1;
        powed[0][0] = powed[1][0] = powed[2][0] = 1;
        for (int i = 1; i < 2 * n; i++)
            for (int j = 0; j < 3; j++) {
                hashes[j][i] = (hashes[j][i - 1] * k[j] + doubled_s[i] -
'0' + 1) % mod[j];
                powed[j][i] = powed[j][i - 1] * k[j] % mod[j];
            }
        vector<int> orig = cut(t);
        for (int i = 1; i <= n; i++) {
            auto cur = get_substr(i, i + n - 1);
            if (cur == orig) {
                cout << "YES";
                return;
            }
        }
        cout << "NO";
    }
}

signed main() {
    solve();
}


//30536769
//21975361
//473
```

Task D

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
```

```cpp
#include <map>
#include <unordered_map>
#include <cassert>

using namespace std;

#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

const int INF = 1e18, N = 2e5 + 228;

inline void ask(char& a) {
    cin >> a;
    if (a == '!')
        exit(0);
}

void solve() {
    int n;
    cin >> n;
    bool flag = true;
    char res;
    while (flag) {
        flag = false;
        int cnt = n - 1;
        while (cnt > 0) {
            cout << "1 1" << endl;
            ask(res);
            cnt--;
            if (res == '<')
                continue;
            flag = true;
            cout << 1 << " " << n << endl;
            cnt--;
            ask(res);
        }
        if (cnt == 0) {
            cout << "1 1" << endl;
            ask(res);
        }
    }
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```

Task E

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <cassert>

using namespace std;

#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

const int N = 1e5 + 228;

vector<int> arr[N];
int pointers[N];

vector<int> path;

void dfs(int v) {
    if (sz(path) > (int)1e5)
        return;
    path.push_back(v);
    int prev = pointers[v];
    pointers[v]++;
    if (pointers[v] == sz(arr[v]))
        pointers[v] = 0;
    dfs(arr[v][prev]);
}

void solve() {
    int n, q;
    cin >> n >> q;
    int t;
    for (int i = 0; i < n; i++) {
        cin >> t;
```

```
        arr[i].resize(t);
        for (int j = 0; j < t; j++) {
            cin >> arr[i][j];
            arr[i][j]--;
        }
    }
    dfs(0);
    int cur;
    for (int i = 0; i < q; i++) {
        cin >> cur;
        cout << path[cur] + 1 << " ";
    }
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```

Task F

```
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("avx2,popcnt,lzcnt")
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <cassert>

using namespace std;

#define sz(v) (int)v.size()
#define walk(v) v.begin(), v.end()

typedef long long ll;
typedef pair<int, int> pii;
typedef long double ld;

const int N = 100;

int arr[N][N];
char aa1[N][N], aa2[N][N], aa3[N][N];
```

```
void solve() {
    int h, w;
    cin >> h >> w;
    char cur;
    for (int i = 0; i < h; i++)
        for (int j = 0; j < w; j++) {
            cin >> cur;
            arr[i][j] = cur - '0';
            aa1[i][j] = aa2[i][j] = aa3[i][j] = '.';
        }

    int ans1 = 0;
    for (int i = 0; i < h; i++) {
        for (int j = 0; j + 1 < w;) {
            ans1 += arr[i][j] * arr[i][j + 1];
            aa1[i][j] = 'L';
            aa1[i][j + 1] = 'R';
            j += 2;
        }
    }
    int ans2 = 0;
    for (int j = 0; j < w; j++) {
        for (int i = 0; i + 1 < h;) {
            aa2[i][j] = 'T';
            aa2[i + 1][j] = 'B';
            ans2 += arr[i][j] * arr[i + 1][j];
            i += 2;
        }
    }
    if (h & 1) {
        for (int j = 0; j + 1 < w;) {
            aa2[h - 1][j] = 'L';
            aa2[h - 1][j + 1] = 'R';
            ans2 += arr[h - 1][j] * arr[h - 1][j + 1];
            j += 2;
        }
    }
    if (w & 1) {
        for (int i = 0; i + 1 < h;) {
            aa1[w - 1][i] = 'T';
            aa1[w - 1][i + 1] = 'B';
            ans1 += arr[w - 1][i] * arr[w - 1][i + 1];
            i += 2;
        }
    }
    int ans3 = 0;
    for (int i = 0; i + 1 < h; i += 2) {
        for (int j = 0; j + 1 < w; j += 2) {
            if (arr[i][j] * arr[i][j + 1] + arr[i + 1][j] * arr[i + 1][j
    + 1] >
                arr[i][j] * arr[i + 1][j] + arr[i][j + 1] * arr[i + 1][j
    + 1]) {
```

```
            aa3[i][j] = 'L';
            aa3[i][j + 1] = 'R';
            aa3[i + 1][j] = 'L';
            aa3[i + 1][j + 1] = 'R';
            ans3 += arr[i][j] * arr[i][j + 1] + arr[i + 1][j] * arr[i
 + 1][j + 1];
        } else {
            aa3[i][j] = 'T';
            aa3[i + 1][j] = 'B';
            aa3[i][j + 1] = 'T';
            aa3[i + 1][j + 1] = 'B';
            ans3 += arr[i][j] * arr[i + 1][j] + arr[i][j + 1] * arr[i
 + 1][j + 1];
        }
    }
}
if (h & 1) {
    for (int j = 0; j + 1 < w;) {
        aa3[h - 1][j] = 'L';
        aa3[h - 1][j + 1] = 'R';
        ans3 += arr[h - 1][j] * arr[h - 1][j + 1];
        j += 2;
    }
}
if (w & 1) {
    for (int i = 0; i + 1 < h;) {
        aa3[w - 1][i] = 'T';
        aa3[w - 1][i + 1] = 'B';
        ans3 += arr[w - 1][i] * arr[w - 1][i + 1];
        i += 2;
    }
}
int fin = max(ans1, max(ans2, ans2));
cout << fin << "\n";
if (ans1 == fin)
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << aa1[i][j];
        }
        cout << "\n";
    }
else if (ans2 == fin)
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << aa2[i][j];
        }
        cout << "\n";
    }
else
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << aa3[i][j];
        }
```

```
            cout << "\n";
        }
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```