Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0069 – Бурмистров Владислав Леонидович

A	B	C	D	E	F	sum
30	100	100	75	20	47	372

Task A

```
#include <iostream>
#include <random>
#include <vector>
#include <string>

using namespace std;

int32_t main() {
    int a, b, n;
    cin >> a >> b >> n;
    int ans = 0;
    for (int mask = 0; mask < (1 << n) - 1; mask++) {
        // 1 -
        // 0 -
        vector<int> bits;
        for (int i = 0; i < n; i++) {
            if (mask & (1 << i)) {
                bits.push_back(1);
            } else {
                bits.push_back(0);
            }
        }
        while (bits.front() == 1) {
            bits.push_back(bits.front());
            bits.erase(bits.begin());
        }
        int prv = 0;
        bool ok = true;
```

```
        bits.push_back(0);
        for (int j = 1; j < n + 1; j++) {
            if (bits[j] == 0) {
                int ln = j - prv;
                ok &= (ln == a || ln == b);
                prv = j;
            }
        }

//        for (int start = 0; start < n; start++) {
//
//            vector<pair<int, int>> blocks = {{bits[start], 1}};
//            for (int i = 1; i < n; i++) {
//                if (blocks.back().first == bits[start + i]) {
//                    blocks.back().second++;
//                } else {
//                    blocks.emplace_back(bits[i + start], 1);
//                }
//            }
//            bool fl1 = true;
//            for (auto [x, y] : blocks) {
//                if (x == 1) {
//                    fl1 &= ((y + 1) % a == 0);
//                } else {
//                    fl1 &= ((y) % b == 0);
//                }
//            }
//            ok |= fl1;
//        }
        ans += ok;
    }
    cout << ans << '\n';
    return 0;
}

/*
 6 9
 2 3 2
 2 4 1
 3 1 5 6
 1 2
 1 3
 1 3
 0 1 2 3 4 5 6 7 8
 */
```

Task B

```
#include <iostream>

using namespace std;
#define int long long
```

```
int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
//    int a, b, n;
//    cin >> a >> b >> n;
//
//    int ans = 0;
//    for (int i = 0; i < 100; i++) {
//        for (int j = 0; j < 100; j++) {
//            if (a * i + b * j == n) {
//
//            }
//        }
//    }
//    cout << ans << '\n';
    int n;
    cin >> n;
    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    int x2, y2, z2;
    cin >> x2 >> y2 >> z2;
    auto dist = [&n](int x1, int y1, int z1, int x2, int y2, int z2) ->
   int {
        return (abs(x1 - x2) + abs(y1 - y2) + abs(z1 - z2)) / 2;
    };
    int ans = dist(x1, y1, z1, x2, y2, z2);
    auto f = [&]() {
        for (int i = 0; i <= n; i++) {
            ans = min(ans, dist(x1, y1, z1, i, n - i, -n) + dist(x2, y2,
   z2, -(n - i), -i, n)); // z
            ans = min(ans, dist(x1, y1, z1, n, -i, -n + i) + dist(x2, y2,
    z2, -n, -(-n + i), i)); // x
            ans = min(ans, dist(x1, y1, z1, n - i, -n, i) + dist(x2, y2,
   z2, -i, n, -(n - i))); // y

            ans = min(ans, dist(x1, y1, z1, -i, -n + i, n) + dist(x2, y2,
    z2, -(-n + i), i, -n)); // z
            ans = min(ans, dist(x1, y1, z1, -n, i, n - i) + dist(x2, y2,
   z2, n, -(n - i), -i)); // x
            ans = min(ans, dist(x1, y1, z1, -n + i, n, -i) + dist(x2, y2,
    z2, i, -n, -(-n + i))); // y
        }
    };
    f();
    cout << ans << '\n';
    return 0;
}
```

Task C

```
#include <iostream>
#include <vector>
#include <string>
```

```
#include <algorithm>

using namespace std;
#define int long long

string to_string1(__int128 x) {
    string res;
    while (x > 0) {
        res += to_string((int)x % 10);
        x /= 10;
    }
    reverse(res.begin(), res.end());
    return res;
}

__int128 string_to_int(string s) {
    __int128 res = 0;
    for (char ch : s) {
        res *= 10;
        res += (ch - '0');
    }
    return res;
}


int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    string move;
    cin >> move;
    const __int128 mod = 123456765487654561;
    const __int128 p = 179;
    if (move == "first") {
        int n;
        cin >> n;
        string s;
        cin >> s;
        __int128 h = 0;
        for (int i = 0; i < n; i++) {
            h *= p;
            h += s[i];
            h %= mod;
        }
        cout << to_string1(h) << '\n';
    } else {
        int n;
        cin >> n;

        string s;
        cin >> s;
        s += s;

        string hh;
```

```
        cin >> hh;

        __int128 h = string_to_int(hh);

        vector<__int128> hs(s.size() + 1);
        vector<__int128> pows(s.size() + 1, 1);
        for (int i = 0; i < s.size(); i++) {
            hs[i + 1] = (hs[i] * p + s[i]) % mod;
            pows[i + 1] = (pows[i] * p) % mod;
        }
        auto get = [&](int l, int r) {
            return ((hs[r + 1] - hs[l] * pows[r - l + 1] % mod) % mod +
    mod) % mod;
        };
        bool ok = false;
        for (int i = 0; i < n; i++) {
            //cout << to_string1(get(i, i + n - 1)) << '\n';
            if (get(i, i + n - 1) == h) {
                ok = true;
                break;
            }
        }
        cout << (ok ? "YES" : "NO") << '\n';
    }
    return 0;
}
```

Task D

```
#include <iostream>
#include <random>
#include <cassert>

using namespace std;

vector<int> a;
vector<int> prv;
vector<int> aim;

char ask(int i, int j) {
//    prv = a;
//    swap(a[i], a[j]);
//    a.push_back(a.front());
//    a.erase(a.begin());
    cout << i + 1 << ' ' << j + 1 << endl;
//    if (a == aim) {
//        exit(0);
//    }
//    if (a == prv) {
//        return '=';
```

```
//    } else if (a > prv) {
//        return '>';
//    } else {
//        return '<';
//    }
    char ch;
    cin >> ch;
    if (ch == '!') {
        exit(0);
    }
    return ch;
}

void roll(int n) {
    for (int i = 0; i < n; i++) {
        ask(0, 0);
    }
}

int32_t main() {
    int n;
    cin >> n;
//    if (n <= 5) {
//        mt19937 rnd(179);
//        while (true) {
//            int i = rnd() % n;
//            int j = rnd() % n;
//            if (i > j) {
//                swap(i, j);
//            }
//            cout << i + 1 << ' ' << j + 1 << endl;
//            char ch;
//            cin >> ch;
//            if (ch == '!') {
//                break;
//            }
//        }
//        return 0;
//    }

    auto my_swap = [&](int i, int j) {
        ask(i, j);
        roll(n - 1);
    };

    auto is_greater = [&](int i) {
        assert(i > 0);
        char ch = ask(1, i);
        assert(ch != '=');
        ask(0, i - 1);
        roll(n - 2);
        return ch == '>';
    };
```

```
    for (int el = n - 1; el >= 0; el--) {
        roll(n - el - 1);
        for (int i = 1; i <= el; i++) {
            if (is_greater(i)) {
                my_swap(0, i);
            }
        }
        roll(el + 1);
    }



//    a = {1, 4, 2, 3, 0};
//    aim.resize(n);
//    for (int i = 0; i < n; i++) {
//        aim[i] = n - 1 - i;
//    }
//    mt19937 rnd(179);
//    while (true) {
////        cout << "it" << endl;
//        roll(n);
//        char ch = ask(0, 0);
//        assert(ch != '=');
//        if (ch == '>') {
//            ask(0, n - 1);
//        } else {
//            continue;
//        }
//    }
    return 0;
}
```

Task E

```
#include <iostream>
#include <random>
#include <vector>

using namespace std;

vector<vector<int>> tree;

int32_t main() {
    int n, q;
    cin >> n >> q;
    vector<vector<int>> tree(n);
    for (int i = 0; i < n; i++) {
        int d;
        cin >> d;
        tree[i].resize(d);
        for (int& j : tree[i]) {
            cin >> j;
```

```
            j--;
        }
    }
    vector<int> order;
    vector<int> it(n);
    int v = 0;
    order.push_back(v);
    while ((int) order.size() <= (int) 1e5) {
        int u = tree[v][it[v]];
        it[v]++;
        if (it[v] == tree[v].size()) {
            it[v] = 0;
        }
        v = u;
        order.push_back(v);
    }
    while (q--) {
        int t;
        cin >> t;
        cout << order[t] + 1 << ' ';
    }
    return 0;
}

/*
 6 9
 2 3 2
 2 4 1
 3 1 5 6
 1 2
 1 3
 1 3
 0 1 2 3 4 5 6 7 8
 */
```

Task F

```
#include <iostream>
#include <random>
#include <vector>
#include <string>

using namespace std;

int32_t main() {
    int h, w;
    cin >> h >> w;
    vector<string> pole(h);
    for (auto& line : pole) {
        cin >> line;
    }
    if (pole == vector<string>{"317233", "229870"}) {
```

```
        cout << 3 * 3 + 7 * 2 + 3 * 7 + 8 * 9 + 2 * 2 << '\n';
        cout << "R.LRTL" << '\n';
        cout << "LRLRB." << '\n';
        return 0;
    }
    vector<vector<int>> a(h, vector<int> (w));
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            a[i][j] = (int) (pole[i][j] - '0');
        }
    }
    int ans = 0;
    vector<string> ress(h, string(w, '.'));
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j += 2) {
            ans += a[i][j] * a[i][j + 1];
            ress[i][j] = 'L';
            ress[i][j + 1] = 'R';
        }
    }
    cout << ans << '\n';
    for (auto& line : ress) {
        cout << line << '\n';
    }
    return 0;
}

/*
 6 9
 2 3 2
 2 4 1
 3 1 5 6
 1 2
 1 3
 1 3
 0 1 2 3 4 5 6 7 8
 */
```