Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0076 – Ваничев Евгений Викторович

A	B	C	D	E	F	sum
100	100	40	100	20	91	451

Task A

```
#include <iostream>

using namespace std;

#define int long long
int memo[100];
int a, b, n;
int count_sp(int rest) {
    if (rest < 0)
        return 0;
    if (memo[rest] != -1)
        return memo[rest];
    return memo[rest] = count_sp(rest - a) + count_sp(rest - b);
}

signed main() {
    cin >> a >> b >> n;
    fill(memo, memo+100, -1);
    memo[0] = 1;
    cout << count_sp(n - a) * a + count_sp(n - b) * b;
    return 0;
}
```

Task B

```
#include <iostream>

using namespace std;
```

```
struct point {
    int x=0;
    int y=0;
    int z=0;
    point(int x, int y, int z) {
        this->x = x;
        this->y = y;
        this->z = z;
    }
    friend istream& operator >> (istream& in, point& pt) {
        in >> pt.x >> pt.y >> pt.z;
        return in;
    }
};

point invert(int n, point pt, char ax) {
    if (ax == 'x') {
        if (pt.x < 0)
            return point(2 * n + pt.x, pt.y - n, pt.z - n);
        return point(pt.x - 2 * n, pt.y + n, pt.z + n);
    }
    if (ax == 'y') {
        if (pt.y < 0)
            return point(pt.x - n, pt.y + 2 * n, pt.z - n);
        return point(pt.x + n, pt.y - 2 * n, pt.z + n);
    }
    if (pt.z < 0)
        return point(pt.x - n, pt.y - n, pt.z + 2 * n);
    return point(pt.x + n, pt.y + n, pt.z - 2 * n);
}

int dist(point a, point b) {
    return (abs(a.x - b.x) + abs(a.y - b.y) + abs(a.z - b.z)) / 2;
}

int main() {
    int n; cin >> n;
    point A{0,0,0}, B{0,0,0}, Bx{0,0,0}, By{0,0,0}, Bz{0,0,0};
    cin >> A >> B;
    Bx = invert(n, B, 'x');
    By = invert(n, B, 'y');
    Bz = invert(n, B, 'z');

    int d = dist(A, B);
    d = min(d, dist(A, Bx));
    d = min(d, dist(A, By));
    d = min(d, dist(A, Bz));
    cout << d;
    return 0;

}
```

Task C

```
time = input()
if (time == "first"):
    input()
    print(input())
else:
    n = int(input())
    s1 = input()
    s2 = input() * 2
    print("YES" if s1 in s2 else "NO")
```

Task D

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

using namespace std;

int main() {
    int n; cin >> n;
    char x;
    int op = n-1;
    cout << "1 1\n";
    cout.flush();
    while (true) {
        cin >> x;
        if (x == '!')
            return 0;
        if (x == '<') {
            cout << "1 1\n";
            cout.flush();
        }
        else {
            cout << "1 " << n << "\n";
            cout.flush();
        }
        --op;
        if (op == 0) {
            op = n - 1;
            cin >> x;
            cout << "1 1\n";
            cout.flush();
        }
    }

}
```

Task E

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

```cpp
#include <map>
#include <set>

using namespace std;
#define int long long

map<int,int> t;
vector<int> delta(1e5+5, 0);
vector<vector<int>> G(1e5+5);
set<int> s;
int Timer;
int C;

int get_next(int v) {
    int n = G[v].size();
    return G[v][(delta[v]++) % n];
}

void dfs(int v) {
    //cout << Timer << " " << C << " " << v+1 << " " << "\n";
    t[Timer++] = v;
    if (Timer > C)
        return;
    dfs(get_next(v));
}

void dfs_10(int v) {
    //cout << Timer << " " << C << " " << v+1 << " " << "\n";
    if (s.count(Timer) != 0)
        t[Timer] = v;
    Timer++;
    if (Timer > C)
        return;
    dfs(get_next(v));
}

signed main() {
    int n, q; cin >> n >> q;
    for (int i = 0; i < n; ++i) {
        int sz; cin >> sz;
        G[i].resize(sz);
        for (int j = 0; j < sz; ++j) {
            cin >> G[i][j];
            G[i][j]--;
        }
    }
    C = 0;
    Timer = 0;
    dfs(0);
    if (q <= 60) {
        vector<int> Q(q);
        for (auto& x: Q) {
            cin >> x;
```

```
            s.insert(-x);
        }
        Timer = 0;
        C = -(*(s.begin()));
        dfs_10(0);
        for (int x: Q)
            cout << t[x] + 1 << " ";
        return 0;
    }
    while (q--) {
        int ti; cin >> ti;
        if (C >= ti)
            cout << t[ti] + 1 << " ";
        else {
            int temp = get_next(t[C]);
            C = ti;
            dfs(temp);
            cout << t[ti] + 1 << " ";
        }
    }
}
```

Task F

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

using namespace std;

vector<vector<int>> board;
vector<vector<char>> answer;

int h, w;

int mod(int a, int mod) {
    return ((a % mod) + mod) % mod;
}

int get(int x, int y) {
    x = mod(x, h);
    y = mod(y, w);
    if (answer[x][y] != '.')
        return 0;
    return board[x][y];
}

void put(int x, int y, char v) {
    x = mod(x, h);
    y = mod(y, w);
    answer[x][y] = v;
}

int get_max_neighboor(int x, int y) {
```

```
    int mx = get(x+1, y);
    mx = max(mx, get(x-1, y));
    mx = max(mx, get(x, y+1));
    return max(mx, get(x, y-1));
}

int get_pr(int x, int y) {
    return get(x, y) * get_max_neighboor(x, y);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cin >> h >> w;
    board.resize(h, vector<int>(w));
    answer.resize(h, vector<char>(w, '.'));
    for (int i = 0; i < h; ++i) {
        string s; cin >> s;
        for (int j = 0; j < w; ++j) {
            board[i][j] = s[j] - '0';
        }
    }
    int S = 0;
    while (true) {
        int im=-1, jm=-1, pm=-1;
        for (int i = 0; i < h; ++i) {
            for (int j = 0; j < w; ++j) {
                if (answer[i][j] != '.') continue;
                int pr = get_pr(i, j);
                if (pm == -1 || pm < pr) {
                    im = i;
                    jm = j;
                    pm = pr;
                }
            }
        }
        if (pm <= 0)
            break;
        S += pm;
        int zn = pm / get(im, jm);

        if (get(im + 1, jm) == zn){
            put(im+1, jm, 'B');
            put(im, jm, 'T');
        }
        else if (get(im-1, jm) == zn) {
            put(im-1, jm, 'T');
            put(im, jm, 'B');
        }
        else if (get(im, jm+1) == zn) {
            put(im, jm+1, 'R');
            put(im, jm, 'L');
        }
```

```
        else {
            put(im, jm-1, 'L');
            put(im, jm, 'R');
        }

    }
    cout << S << "\n";
    for (auto line: answer){
        for (auto c: line)
            cout << c;
        cout << "\n";
    }
}
```