Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0519 – Ярыкин Евгений Павлович

A	B	C	D	E	F	sum
100	50	40	100	20	90	400

Task A

```
print(1)
```

Task B

```
#include <iostream>

using namespace std;

#define RANGE(i, n) for (int i = 0; i < n; i++)
#define LOOP(n) RANGE(_, n)
#define LOOP2(n) RANGE(__, n)
#define elif else if

#ifdef LOCAL
#define ENDLINE endl
#define FLUSH flush
#define DEBUG(x) cerr << #x << ": " << x << endl;
#else
#define FLUSH ""
#define ENDLINE '\n'
#define DEBUG(x) ;
#endif

template<typename T>
T max(T &a, T &b, T&c) {
    return max(max(a, b), c);
}

struct Point {
```

```
    int x, y, z;

    Point(int x, int y, int z) {
        this->x = x;
        this->y = y;
        this->z = z;
    }

    Point() {
        this->x = this->y = this->z = 0;
    }


    int len(Point &p) {
        int x = abs(this->x - p.x),
                y = abs(this->y - p.y),
                z = abs(this->z - p.z);
        return x + y + z - max(x, y, z);
    }
};

istream &operator>>(istream &in, Point &p) {
    in >> p.x >> p.y >> p.z;
    return in;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n;
    Point p1, p2;
    cin >> n >> p1 >> p2;
    cout << p1.len(p2) << ENDLINE;
}
```

Task C

```
#include <iostream>

using namespace std;

#define RANGE(i, n) for (int i = 0; i < n; i++)
#define LOOP(n) RANGE(_, n)
#define LOOP2(n) RANGE(__, n)
#define elif else if

#ifdef LOCAL
#define ENDLINE endl
#define FLUSH flush
#define DEBUG(x) cerr << #x << ": " << x << endl;
#else
#define FLUSH ""
```

```
#define ENDLINE '\n'
#define DEBUG(x) ;
#endif

using ll = long long;

template<typename T>
T max(T &a, T &b, T &c) {
    return max(max(a, b), c);
}

template<typename T>
T min(T a, T b, T c) {
    return min(min(a, b), c);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    string s;
    cin >> s;
    if (s[0] == 'f') {
        int t;
        cin >> t;
        cin >> s;
        cout << s;
    } else {
        int t;
        string s1, s2;
        cin >> t >> s1 >> s2;
        RANGE(i, t) {
            bool ok = true;
            RANGE(j, t) {
                if (s1[j] != s2[(j + i) % t]) {
                    ok = false;
                    break;
                }
            }
            if (ok) {
                cout << "YES" << ENDLINE;
                return 0;
            }
        }
        cout << "NO" << ENDLINE;
    }
}
```

Task D

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

```
using namespace std;

#define RANGE(i, n) for (int i = 0; i < n; i++)
#define LOOP(n) RANGE(_, n)
#define LOOP2(n) RANGE(__, n)
#define elif else if

#ifdef LOCAL
#define ENDLINE endl
#define FLUSH flush
#define DEBUG(x) cerr << #x << ": " << x << endl;
#else
#define FLUSH ""
#define ENDLINE '\n'
#define DEBUG(x) ;
#endif

using ll = long long;
using pii = pair<int, int>;

template<typename T>
T max(T &a, T &b, T &c) {
    return max(max(a, b), c);
}

template<typename T>
T max(vector<T> &v) {
    T _max = v[0];
    for (auto &val: v) {
        _max = max(_max, val);
    }
    return _max;
}

template<typename T>
T min(T a, T b, T c) {
    return min(min(a, b), c);
}

template<typename T>
istream &operator>>(istream &in, vector<T> &v) {
    for (auto &val: v) {
        in >> val;
    }
    return in;
}

int n, k = 0;
vector<int> my;
int sorts = 1;

int _top(int i) {
    return ((i - k) % n + n) % n + 1;
```

```
}

char moveii(int i, int j) {
    cout << _top(i) << " " << _top(j) << endl;
    swap(my[i], my[j]);
    k++;
    char ch;
    cin >> ch;
    if (ch == '!') {
        exit(0);
    }
    return ch;
}

char moveia(int i, int j) {
    return moveii(i, (j + k) % n );
}

char moveai(int i, int j) {
    return moveii((i + k) % n, j);
}


void insert(int i, int to_i) {
    while (i > to_i) {
        moveii(i, i - 1);
        i--;
    }
}

int index(int val) {
    RANGE(i, n) {
        if (my[i] == val) {
            return i;
        }
    }
}

void sort(int i) {
    int L = -1, R = i;
    while (L + 1 < R) {
        int c = (L + R) / 2;
        moveia(c, 1);
        char x = moveia(index(i), 1);
        moveai(n - 1, c);
        moveii(index(i), i);
        k %= n;
        if (x == '>') {
            R = c;
        } else {
            L = c;
        }
    }
```

```
    insert(i, R);
    sorts++;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    cin >> n;
    my.resize(n);
    RANGE(i, n) {
        my[i] = i;
    }
    while (sorts < n) {
        sort(sorts);
    }
    while (true) {
        cout << "1 1" << endl;
        char t;
        cin >> t;
        if (t == '!') {
            return 0;
        }
    }
}
```

Task E

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

using namespace std;

#define RANGE(i, n) for (int i = 0; i < n; i++)
#define LOOP(n) RANGE(_, n)
#define LOOP2(n) RANGE(__, n)
#define elif else if

#ifdef LOCAL
#define ENDLINE endl
#define FLUSH flush
#define DEBUG(x) cerr << #x << ": " << x << endl;
#else
#define FLUSH ""
#define ENDLINE '\n'
#define DEBUG(x) ;
#endif

using ll = long long;

template<typename T>
T max(T &a, T &b, T &c) {
    return max(max(a, b), c);
```

```
}

template<typename T>
T min(T a, T b, T c) {
    return min(min(a, b), c);
}

template<typename T>
istream &operator>>(istream &in, vector<T> &v) {
    for (auto &val: v) {
        in >> val;
    }
    return in;
}

vector<vector<int>> g;
vector<int> p, sm, full_time, chile_count_ok;
vector<bool> this_ok, this_full_ok;
int t_now = 0;
int v_now = 0;

inline int get(int i, int j) {
    auto &gi = g[i];
    if (j < 0) {
        return gi[j + gi.size()];
    } elif (j >= gi.size()) {
        return gi[j % gi.size()];
    } else {
        return gi[j];
    }
}

void dfs(int v, int from) {
    p[v] = from;
    for (int u: g[v]) {
        if (u != from) {
            dfs(u, v);
            /*this_full_ok[v] = this_full_ok[v] && this_full_ok[u];
            full_time[v] += 2 + full_time[u];
            chile_count_ok[v] += this_full_ok[u];*/
        }
    }
    /*if (g[v].back() == from) {
        this_ok[v] = true;
    } else {
        this_full_ok[v] = false;
    }*/
}

void run_to(int t) {
    while (t_now < t) {
        int v = g[v_now][sm[v_now]];
        sm[v_now]++;
```

```
        if (sm[v_now] == g[v_now].size()) {
            sm[v_now] = 0;
        }
        v_now = v;
        t_now++;
    }
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n, q;
    cin >> n >> q;
    g.resize(n);
    p.resize(n);
    sm.resize(n, 0);
    RANGE(i, n) {
        int t;
        cin >> t;
        auto &gi = g[i];
        gi.resize(t);
        RANGE(j, t) {
            cin >> gi[j];
            gi[j]--;
        }
    }
    dfs(0, 0);
    LOOP(q) {
        int t;
        cin >> t;
        run_to(t);
        cout << v_now + 1 << " " << FLUSH;
    }
    cout << ENDLINE;
}
```

Task F

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

using namespace std;

#define RANGE(i, n) for (int i = 0; i < n; i++)
#define LOOP(n) RANGE(_, n)
#define LOOP2(n) RANGE(__, n)
#define elif else if

#ifdef LOCAL
#define ENDLINE endl
#define FLUSH flush
```

```
#define DEBUG(x) cerr << #x << ": " << x << endl;
#else
#define FLUSH ""
#define ENDLINE '\n'
#define DEBUG(x) ;
#endif

using ll = long long;
using pii = pair<int, int>;

template<typename T>
T max(T &a, T &b, T &c) {
    return max(max(a, b), c);
}

template<typename T>
T max(vector<T> &v) {
    T _max = v[0];
    for (auto &val: v) {
        _max = max(_max, val);
    }
    return _max;
}

template<typename T>
T min(T a, T b, T c) {
    return min(min(a, b), c);
}

template<typename T>
istream &operator>>(istream &in, vector<T> &v) {
    for (auto &val: v) {
        in >> val;
    }
    return in;
}

int w, h;
vector<vector<int>> s;

namespace solve1_1 {
    int get_ans() {
        int ans = 0;
        for (int i = 0; i < h; i += 2) {
            RANGE(j, w) {
                ans += s[i][j] * s[i + 1][j];
            }
        }
        return ans;
    }

    void print() {
        LOOP(h / 2) {
```

```
            LOOP2(w) {
                cout << 'T';
            }
            cout << ENDLINE;
            LOOP2(w) {
                cout << 'B';
            }
            cout << ENDLINE;
        }
    }
}

namespace solve1_2 {
    int get_ans() {
        int ans = 0;
        for (int i = 1; i < h - 1; i += 2) {
            RANGE(j, w) {
                ans += s[i][j] * s[i + 1][j];
            }
        }
        RANGE(j, w) {
            ans += s[0][j] * s[h - 1][j];
        }
        return ans;
    }

    void print() {
        LOOP(h / 2) {
            LOOP2(w) {
                cout << 'B';
            }
            cout << ENDLINE;
            LOOP2(w) {
                cout << 'T';
            }
            cout << ENDLINE;
        }
    }
}

namespace solve2_1 {
    int get_ans() {
        int ans = 0;
        RANGE(i, h) {
            for (int j = 0; j < w; j += 2) {
                ans += s[i][j] * s[i][j + 1];
            }
        }
        return ans;
    }

    void print() {
        LOOP(h) {
```

```
            LOOP2(w / 2) {
                cout << "LR";
            }
            cout << ENDLINE;
        }
    }
}

namespace solve2_2 {
    int get_ans() {
        int ans = 0;
        RANGE(i, h) {
            for (int j = 1; j < w - 1; j += 2) {
                ans += s[i][j] * s[i][j + 1];
            }
            ans += s[i][0] * s[i][w - 1];
        }
        return ans;
    }

    void print() {
        LOOP(h) {
            LOOP2(w / 2) {
                cout << "RL";
            }
            cout << ENDLINE;
        }
    }
}

namespace solve3 {
    vector<vector<char>> p_ans;

    inline void normal(int &i, int &j) {
        if (i == -1) {
            i = h - 1;
        } elif (i == h) {
            i = 0;
        }
        if (j == -1) {
            j = w - 1;
        } elif (j == w) {
            j = 0;
        }
    }

    int get(int i, int j) {
        normal(i, j);
        if (p_ans[i][j] != '.') {
            return 0;
        }
        return s[i][j];
    }
```

```
void set(int i, int j, char c) {
    normal(i, j);
    p_ans[i][j] = c;
}

int get_ans() {
    p_ans.resize(h, vector<char>(w, '.'));
    int ans = 0;
    while (true) {
        pii bestij = {-1, -1},
                napr = {0, 0};
        int bestp = 0;
        RANGE(i, h) {
            RANGE(j, w) {
                int v = get(i, j);
                if (v) {
                    if (bestp < v * get(i, j + 1)) {
                        bestij = {i, j};
                        napr = {0, 1};
                        bestp = v * get(i, j + 1);
                    }
                    if (bestp < v * get(i, j - 1)) {
                        bestij = {i, j - 1};
                        napr = {0, 1};
                        bestp = v * get(i, j - 1);
                    }
                    if (bestp < v * get(i + 1, j)) {
                        bestij = {i, j};
                        napr = {1, 0};
                        bestp = v * get(i + 1, j);
                    }
                    if (bestp < v * get(i - 1, j)) {
                        bestij = {i - 1, j};
                        napr = {1, 0};
                        bestp = v * get(i - 1, j);
                    }
                }
            }
        }
        if (bestp) {
            if (napr.first == 1) {
                set(bestij.first, bestij.second, 'T');
                set(bestij.first + 1, bestij.second, 'B');
            } else {
                set(bestij.first, bestij.second, 'L');
                set(bestij.first, bestij.second + 1, 'R');
            }
            ans += bestp;
        } else {
            break;
        }
    }
```

```
        return ans;
    }

    void print() {
        RANGE(i, h) {
            RANGE(j, w) {
                cout << p_ans[i][j];
            }
            cout << ENDLINE;
        }
    }
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    cin >> h >> w;
    s.resize(h, vector<int>(w));
    RANGE(i, h) {
        RANGE(j, w) {
            char t;
            cin >> t;
            s[i][j] = t - '0';
        }
    }
    /*cout << solve3::get_ans() << ENDLINE;
    solve3::print();*/
    vector<void (*)()> f = {solve1_1::print, solve1_2::print, solve2_1::
   print, solve2_2::print, solve3::print};
    vector<int> anses = {solve1_1::get_ans(), solve1_2::get_ans(),
   solve2_1::get_ans(), solve2_2::get_ans(),
                         solve3::get_ans()};
    int m = max(anses);
    cout << m << ENDLINE;
    RANGE(i, f.size()) {
        if (m == anses[i]) {
            f[i]();
            break;
        }
    }
}
```