Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0461 – Туров Макар Олегович

A	B	C	D	E	F	sum
100	100	100	100	20	38	458

Task A

```
#include <bits/stdc++.h>

#define int int64_t

#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 5e5 + 2;;
const int MAXX = 2e6 + 200;
const int BS = 20;
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
```

```
            a %= b;
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
    return 1;
}

mt19937 rnd(123);

int get(int l, int a, int b){
    if(a < b){
        swap(a, b);
    }
    int ans = 1;
    int cnt = 1;
    for(int i = a + 1; i <= l; i++){
        ans *= i;
        while(cnt <= b && ans % cnt == 0){
            ans /= cnt++;
        }
    }
    return ans;
}

void solve(){
    int n, a, b;
    cin >> a >> b >> n;
    int cnt = 0;
    for(int i = 0; i <= n; i++){
        if(n - a * i >= 0 && (n - a * i) % b == 0){
            int c1 = i;
            int c2 = (n - a * i) / b;
            if(c1){
                cnt += get(c1 + c2 - 1, c1 - 1, c2) * a;
            }
            if(c2){
                cnt += get(c1 + c2 - 1, c1, c2 - 1) * b;
            }
        }
    }
```

```
    cout << cnt;
}

signed main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
    }
    return 0;
}
```

Task B

```
#include <bits/stdc++.h>

#define int int64_t

#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 5e5 + 2;;
const int MAXX = 2e6 + 200;
const int BS = 20;
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
            a %= b;
```

```
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
    return 1;
}

mt19937 rnd(123);

int find_dst(vector <int> a, vector <int> b){
    int cnt = 0;
    for(int i = 0; i < 3; i++){
        if(a[i] - b[i] >= 0){
            cnt += a[i] - b[i];
        }
    }
    return cnt;
}

int n;
vector <int> f;
vector <int> s;

int get(int i, int j){
    vector <int> f1 = f;
    vector <int> s1 = s;
    int ans = INF;
    int add = n - f1[i];
    f1[j] -= add;
    f1[i] += add;
    f1[i] = -f1[i];
    f1[j] = -f1[j];
    f1[3 - i - j] = -f1[3 - i - j];
    swap(f1[j], f1[3 - i - j]);
    ans = min(ans, find_dst(f1, s1) + add);
    f1 = f;
    s1 = s;
    add = n + f1[i];
    f1[j] += add;
    f1[i] -= add;
    f1[i] = -f1[i];
```

```
    f1[j] = -f1[j];
    f1[3 - i - j] = -f1[3 - i - j];
    swap(f1[j], f1[3 - i - j]);
    ans = min(ans, find_dst(f1, s1) + add);
    return ans;
}

int gg(){
    int ans = find_dst(f, s);
    ans = min({ans, get(0, 1), get(1, 0), get(2, 0), get(0, 2), get(1, 2)
   , get(2, 1)});
    return ans;
}

void solve(){
    cin >> n;
    for(int i = 0; i < 1; i++){
        int x, y, z;
        cin >> x >> y >> z;
        f.push_back(x);
        f.push_back(y);
        f.push_back(z);
    }
    for(int i = 0; i < 1; i++){
        int x, y, z;
        cin >> x >> y >> z;
        s.push_back(x);
        s.push_back(y);
        s.push_back(z);
    }
    int a1 = gg();
    swap(f, s);
    a1 = min(a1, gg());
    cout << a1;
}

signed main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
    }
    return 0;
}
```

Task C

```
#include <bits/stdc++.h>

#define int int64_t

//#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 5e5 + 2;;
const int MAXX = 2e6 + 200;
const int BS = 20;
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
            a %= b;
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
    return 1;
}

mt19937 rnd(103);
```

```
vector <int> col(MAXN);
vector <int> col1(MAXN);
vector <int> col2(MAXN);

bool cmp(int a, int b){
    if(col[a] == col[b]){
        return col1[a] < col1[b];
    }
    return col[a] < col[b];
}

string get_s(string s){
    char last = s[0];
    int pos = 0;
    for(int i = 1; i < s.size(); i++){
        if(s[i] != last){
            pos = i;
            break;
        }
    }
    string s1 = "";
    for(int i = pos; i < pos + s.size(); i++){
        s1 += s[i % s.size()];
    }
    swap(s, s1);
    vector <int> v;
    for(int i = 0; i < s.size(); i++){
        col[i] = s[i] - '0';
        v.push_back(i);
    }
    sort(v.begin(), v.end(), cmp);
    for(int i = 0; i < LG; i++){
        if((1ll << i) > s.size()){
            break;
        }
        vector <int> v1;
        int nw;
        int n_col = 0;
        int l = 0;
        while(l != v.size()){
            vector <int> v2;
            int n_b = col[v[l]];
            v2.push_back(v[l]);
            col1[v[l]] = col[(v[l] + (1ll << i)) % s.size()];
            l++;
            while(l != v.size() && col[v[l]] == n_b){
                v2.push_back(v[l]);
                col1[v[l]] = col[(v[l] + (1ll << i)) % s.size()];
                l++;
            }
            sort(v2.begin(), v2.end(), cmp);
            col2[v2[0]] = n_col;
            nw = col1[v2[0]];
```

```
            v1.push_back(v2[0]);
            for(int j = 1; j < v2.size(); j++){
                if(nw != col1[v2[j]]){
                    n_col++;
                    nw = col1[v2[j]];
                }
                col2[v2[j]] = n_col;
                v1.push_back(v2[j]);
            }
            n_col++;
        }
        swap(col2, col);
        swap(v1, v);
    }
    s1 = "";
    for(int i = v[0]; i < v[0] + s.size(); i++){
        s1 += s[i % s.size()];
    }
    return s1;
}

vector <int> pss;

string get_zn(string s){
    string a = "";
    for(int i = 0; i < LG; i++){
        int xr = 0;
        for(int j = 0; j < s.size(); j+= (1 << i)){
            xr ^= (s[j] - '0');
        }
        a += to_string(xr);
    }
    for(auto i:pss){
        a += s[i % s.size()];
    }
    int v = 0;
    for(int i = 0; i < s.size(); i++){
        if(s[i] == '1'){
            v += bp(2, i);
            v %= MOD;
        }
    }
    a += to_string(v);
    return a;
}

void solve(){
    string T;
    cin >> T;
    if(T[0] == 'f'){
        int n;
        cin >> n;
        string s;
```

```
        cin >> s;
        cout << get_zn(get_s(s));
    }
    else{
        int n;
        cin >> n;
        string s;
        cin >> s;
        string gt;
        cin >> gt;
        cout << (gt == get_zn(get_s(s)) ? "YES" : "NO");
    }

}
//
    1010000000000000000000000000000100011011110001000011100000010000001101110101011000101011

signed main(){
    for(int i = 0; i < 50; i++){
        pss.push_back(rnd());
    }
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
    }
    return 0;
}
```

Task D

```
#include <bits/stdc++.h>

#define int int64_t

//#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 5e5 + 2;;
const int MAXX = 2e6 + 200;
const int BS = 20;
```

```
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
            a %= b;
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
    return 1;
}

mt19937 rnd(123);

vector <int> vp[MAXN];

vector <int> pos(MAXN);
vector <int> wh(MAXN);

int del = 0;

int n;

int get_ps(int v){
    return (pos[v] - del + n) % n;
}

void d(int i, int ps){
```

```
    cout << pos[i] + 1 << ' ' << ps + 1 << endl;
}

bool cmp(int a, int b){
    return vp[a].size() < vp[b].size();
}

void solve(){
    cin >> n;
    vector <int> vv3;
    vector <int> vv;
    for(int i = 0; i < n; i++){
        pos[i] = wh[i] = i;
        vv.push_back(i);
        vv3.push_back(i);
    }
    char c;
    for(int i = 0; i < n; i++){
        for(int j = i + 1; j < n; j++){
            d(i, 2);
            swap(vv[pos[i]], vv[2]);
            for(int k = 0; k < n - 1; k++){
                swap(vv[k], vv[k + 1]);
            }
            for(int k = 0; k < n; k++){
                pos[vv[k]] = k;
            }
            cin >> c;
            if(c == '!'){
                return;
            }
            d(j, 2);
            swap(vv[pos[j]], vv[2]);
            for(int k = 0; k < n - 1; k++){
                swap(vv[k], vv[k + 1]);
            }
            for(int k = 0; k < n; k++){
                pos[vv[k]] = k;
            }
            cin >> c;
            if(c == '!'){
                return;
            }
            cout << 1 << ' ' << 1 << endl;
            for(int k = 0; k < n - 1; k++){
                swap(vv[k], vv[k + 1]);
            }
            for(int k = 0; k < n; k++){
                pos[vv[k]] = k;
            }
            cin >> c;
            if(c == '!'){
                return;
```

```
            }
            if(c == '<'){
                vp[i].push_back(j);
            }
            else{
                vp[j].push_back(i);
            }
        }
    }
    sort(vv3.begin(), vv3.end(), cmp);
    reverse(vv3.begin(), vv3.end());
    for(auto i:vv3){
        d(i, n - 1);
        swap(vv[pos[i]], vv[n - 1]);
        for(int k = 0; k < n - 1; k++){
            swap(vv[k], vv[k + 1]);
        }
        for(int k = 0; k < n; k++){
            pos[vv[k]] = k;
        }
        cin >> c;
        if(c == '!'){
            return;
        }
    }
    while(1){
        cout << 1 << ' ' << 1 << endl;
        cin >> c;
        if(c == '!'){
            return;
        }
    }
}

signed main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
    }
    return 0;
}
```

Task E

```
#include <bits/stdc++.h>

#define int int64_t

//#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 1e5 + 5;
const int MAXX = 1e8;
const int MAXXN = 1e8;
const int BS = 20;
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
            a %= b;
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
    return 1;
}

mt19937 rnd(288);
```

```
vector <int> del(MAXN);
vector <int> vp[MAXN];

vector <int> ans(MAXN);

void get(){
    int v = 0;
    int t = 0;
    while(t < MAXN){
        ans[t] = v;
        int v1 = vp[v][del[v] % vp[v].size()];
        del[v]++;
        v = v1;
        t++;
    }
}

void solve(){
    int n, q;
    cin >> n >> q;
    for(int i = 0; i < n; i++){
        int k;
        cin >> k;
        for(int j = 0; j < k; j++){
            int f;
            cin >> f;
            f--;
            vp[i].push_back(f);
        }
    }
    get();
    while(q--){
        int t;
        cin >> t;
        cout << ans[t] + 1 << ' ';
    }
}

signed main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
```

```
    }
    return 0;
}
```

Task F

```
#include <bits/stdc++.h>

#define int int64_t

//#define endl '\n'

using namespace std;

const int INF = 1e15;
const int MAXN = 50;;
const int MAXX = 1e8;
const int MAXXN = 1e8;
const int BS = 20;
const int MOD = 998244353;
const int LG = 30;

int bp(int a, int p){
    if(p == 0){
        return 1;
    }
    if(p & 1){
        return bp(a, p - 1) * a % MOD;
    }
    int k = bp(a, p / 2);
    return k * k % MOD;
}

int gcd(int a, int b){
    while(a && b){
        if(a > b){
            a %= b;
        }
        swap(a, b);
    }
    return a + b;
}

int nok(int a, int b){
    return a / gcd(a, b) * b;
}

bool is_prime(int x){
    for(int i = 2; i * i <= x; i++){
        if(x % i == 0){
            return 0;
        }
    }
```

```
    return 1;
}

mt19937 rnd(112);

long double T = 1;

int ans = 0;

vector <vector<int>> opt;
vector <string> nw;

int n, m;
int sm = 0;
vector <pair<int, int>> go = {{-1, 0}, {1, 0}, {0, -1}, {0, 1}};

vector <vector <int>> v(MAXN, vector <int> (MAXN, -1));
vector <vector<int>> vs;

int kol = 0;

int get(int x, int y){
    for(auto [a, b]:go){
        int x1 = (x + a + n) % n;
        int y1 = (y + b + m) % m;
        if(v[x1][y1] == v[x][y] && v[x][y] != -1){
            return vs[x1][y1] * vs[x][y];
        }
    }
    return 0;
}

void delet(int x, int y){
    for(auto [a, b]:go){
        int x1 = (x + a + n) % n;
        int y1 = (y + b + m) % m;
        if(v[x1][y1] == v[x][y]){
            v[x1][y1] = -1;
        }
    }
    v[x][y] = -1;
}

bool yes(){
    return rnd() % (MAXX + 1) >= MAXX * T;
}

void do_trn(int x, int y, int t){
    int x1 = (x + go[t].first + n) % n;
    int y1 = (y + go[t].second + m) % m;
    if(v[x][y] == v[x1][y1] && v[x][y] != -1){
        return;
    }
```

```
    int del = vs[x][y] * vs[x1][y1] - get(x, y) - get(x1, y1);
    if(del > 0 || yes()){
        delet(x, y);
        delet(x1, y1);
        v[x][y] = kol;
        v[x1][y1] = kol;
        kol++;
        sm += del;
        if(sm > ans){
            ans = sm;
            opt = v;
        }
    }
}

void solve(){
    long double f = clock();
    cin >> n >> m;
    for(int i = 0; i < n; i++){
        string s;
        cin >> s;
        vector <int> v1;
        string s1 = "";
        for(int j = 0; j < m; j++){
            s1 += '.';
            v1.push_back(s[j] - '0');
        }
        nw.push_back(s1);
        vs.push_back(v1);
    }
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            if(j % 2 == 0){
                do_trn(i, j, 3);
            }
        }
    }
    long double del = 0.564;
    while((clock() - f) / CLOCKS_PER_SEC < 1.95){
        T *= del;
        int x = rnd() % n;
        int y = rnd() % m;
        int t = rnd() % 4;
        do_trn(x, y, t);
    }
    cout << ans << endl;
    map <pair<int, int>, char> mp;
    mp[{0, 1}] = 'R';
    mp[{0, -1}] = 'L';
    mp[{-1, 0}] = 'B';
    mp[{1, 0}] = 'T';
    for(int x = 0; x < n; x++){
        for(int y = 0; y < m; y++){
```

```
            if(opt[x][y] == -1){
                cout << nw[x][y];
                continue;
            }
            for(auto [a, b]:go){
                int x1 = (x + a + n) % n;
                int y1 = (y + b + m) % m;
                if(opt[x1][y1] == opt[x][y] && opt[x][y] != -1){
                    //cout << "NOW: " << x << ' ' << y << ' ' << x1 << '
    ' << y1 << endl;
                    nw[x1][y1] = mp[{a, b}];
                    nw[x][y] = mp[{-a, -b}];
                    opt[x1][y1] = -1;
                    opt[x][y] = -1;
                    break;
                }
            }
            cout << nw[x][y];
        }
        cout << endl;
    }
}

signed main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int t = 1;
    //cin >> t;
    long double f = clock();
    while (t--){
        solve();
        long double s = clock();
        #ifdef time
        cout << (s - f) / CLOCKS_PER_SEC;
        #endif
        cout << endl;
    }
    return 0;
}
```