Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0057 – Болуць Роман Валерьевич

A	B	C	D	E	F	sum
100	100	40	35	20	71	366

Task A

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    long long a, b, n; cin >> a >> b >> n;
    long long ans = 0;
    vector <vector<long long>> mas(100, vector <long long>(100, 1));
    for (int i = 1; i < 100; i++) {
        mas[0][i] = 0;
    }
    for (long long i = 1; i < 100; i++) {
        for (long long j = 1; j < 100; j++) {
            mas[i][j] = mas[i - 1][j] + mas[i - 1][j - 1];
        }
    }
```

```
    for (long long i = 0; i <= n; i++) {
        long long ost = n - a * i;
        if (ost >= 0 && ost % b == 0) {
            long long j = ost / b;
            //              b*c(i+j-1,j-1)+a*c(i+j-1,i-1)
            if (i == 0) {
                ans += b;
            }
            else if (j == 0) {
                ans += a;
            }
            else {
                ans += b * mas[i + j - 1][j - 1] + a * mas[i + j - 1][i -
 1];
            }
        }
    }
    cout << ans;
    return 0;
}
```

Task B

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
int res(int x1, int y1, int z1, int x2, int y2, int z2) {
    return max(max(abs(x1 - x2), abs(y1 - y2)), abs(z1 - z2));
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int n; cin >> n;
    int x1, y1, z1, x2, y2, z2; cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
    int ans = res(x1, y1, z1, x2, y2, z2);
    for (int y = -n; y <= 0; y++) {
        //z
        int x = 0 - n - y;
        ans = min(ans, res(x1, y1, z1, x, y, n) + res(-y, -x, -n, x2, y2,
 z2));
        ans = min(ans, res(x1, y1, z1, -x, -y, -n) + res(y, x, n, x2, y2,
```

```
    z2));
    }
    for (int z = -n; z <= 0; z++) {
        //x
        int y = 0 - n - z;
        ans = min(ans, res(x1, y1, z1, n, y, z) + res(-n, -z, -y, x2, y2,
    z2));
        ans = min(ans, res(x1, y1, z1, -n, -y, -z) + res(n, z, y, x2, y2,
    z2));
    }
    for (int x = -n; x <= 0; x++) {
        //y
        int z = 0 - n - x;
        ans = min(ans, res(x1, y1, z1, x, n, z) + res(-z, -n, -x, x2, y2,
    z2));
        ans = min(ans, res(x1, y1, z1, -x, -n, -z) + res(z, n, x, x2, y2,
    z2));
    }
    cout << ans;
    return 0;
}
```

Task C

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
string sf = "111111";
int colvo1(string s, int n) {
    int cnt = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            cnt++;
        }
    }
    return cnt;
}
int maxposl1(string s, int n) {
    int maxcnt = 0, tekcnt = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            tekcnt++;
```

```
        } else {
            maxcnt = max(maxcnt, tekcnt);
            tekcnt = 0;
        }
    }
    maxcnt = max(tekcnt, maxcnt);
    if (maxcnt == n) { return n; }
    if (maxcnt == 0) { return 0; }
    int lind = 0, rind = n - 1, ccn = 0;
    while (s[lind] == '1') { lind++; ccn++; }
    while (s[rind] == '1') { rind--; ccn++; }
    maxcnt = max(maxcnt, ccn);
    return maxcnt;
}
int maxposl0(string s, int n) {
    int maxcnt = 0, tekcnt = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '0') {
            tekcnt++;
        } else {
            maxcnt = max(maxcnt, tekcnt);
            tekcnt = 0;
        }
    }
    maxcnt = max(tekcnt, maxcnt);
    if (maxcnt == n) { return n; }
    if (maxcnt == 0) { return 0; }
    int lind = 0, rind = n - 1, ccn = 0;
    while (s[lind] == '0') { lind++; ccn++; }
    while (s[rind] == '0') { rind--; ccn++; }
    maxcnt = max(maxcnt, ccn);
    return maxcnt;
}
int changes(string s, int n) {
    int cnt = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] != s[(i + 1) % n]) { cnt++; }
    }
    return cnt;
}
long long maxinvs(string s, long long n) {
    long long sumcoor = 0, colvo11 = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            sumcoor += i; colvo11++;
        }
    }
    long long notinv = (n * (n - 1)) / 2 - ((n - colvo11 - 1) * (n -
   colvo11)) / 2;
    long long inv = notinv - sumcoor, maxinv = inv;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            inv -= (n - colvo11);
```

```
        } else {
            inv += colvo11;
            maxinv = max(maxinv, inv);
        }
    }
    return maxinv;
}
long long mininvs(string s, long long n) {
    long long sumcoor = 0, colvo11 = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            sumcoor += i; colvo11++;
        }
    }
    long long notinv = (n * (n - 1)) / 2 - ((n - colvo11 - 1) * (n -
   colvo11)) / 2;
    long long inv = notinv - sumcoor, mininv = inv;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            inv -= (n - colvo11);
            mininv = min(mininv, inv);
        }
        else {
            inv += colvo11;
        }
    }
    return mininv;
}
long long suminvs(string s, long long n) {
    long long sumcoor = 0, colvo11 = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            sumcoor += i; colvo11++;
        }
    }
    long long notinv = (n * (n - 1)) / 2 - ((n - colvo11 - 1) * (n -
   colvo11)) / 2;
    long long inv = notinv - sumcoor, suminv = inv;
    for (int i = 0; i < n - 1; i++) {
        if (s[i] == '1') {
            inv -= (n - colvo11);
            suminv += inv;
        }
        else {
            inv += colvo11;
            suminv += inv;
        }
    }
    return suminv;
}
long long prinvs(string s, long long n) {
    long long sumcoor = 0, colvo11 = 0;
    for (int i = 0; i < n; i++) {
```

```
        if (s[i] == '1') {
            sumcoor += i; colvo11++;
        }
    }
    long long notinv = (n * (n - 1)) / 2 - ((n - colvo11 - 1) * (n -
    colvo11)) / 2;
    long long inv = notinv - sumcoor, prinv = inv;
    for (int i = 0; i < n - 1; i++) {
        if (s[i] == '1') {
            inv -= (n - colvo11);
            prinv = (prinv * inv) % 1000000007;
        }
        else {
            inv += colvo11;
            prinv = (prinv * inv) % 1000000007;
        }
    }
    return prinv;
}
int colvo101(string s, int n) {
    int cnt = 0;
    for (int i = 0; i < n; i++) {
        int nar = 0;
        int gd = sf.size();
        for (int j = 0; j < gd; j++) {
            if (s[(i + j) % n] != sf[j]) { nar++; }
        }
        if (nar == 0) { cnt++; }
    }
    return cnt;
}
long long hashinvs(string s, long long n) {
    long long sumcoor = 0, colvo11 = 0;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            sumcoor += i; colvo11++;
        }
    }
    long long notinv = (n * (n - 1)) / 2 - ((n - colvo11 - 1) * (n -
    colvo11)) / 2;
    long long inv = notinv - sumcoor;
    vector <long long> invs; invs.push_back(inv);
    for (int i = 0; i < n - 1; i++) {
        if (s[i] == '1') {
            inv -= (n - colvo11);
            invs.push_back(inv);
        }
        else {
            inv += colvo11;
            invs.push_back(inv);
        }
    }
    sort(invs.begin(), invs.end());
```

```
    long long p = 31, mod = 1e9 + 13, ress = 0;
    for (auto elem : invs) {
        ress = (ress * p) % mod;
        ress = (ress + elem + 1) % mod;
    }
    return ress;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    string player; cin >> player;
    if (player == "first") {
        int n; cin >> n;
        string s; cin >> s;
        if (n <= 100) {
            cout << s << endl;
        } else { cout << colvo1(s, n) << " " << maxposl1(s, n) << " " <<
    maxposl0(s, n) << " " << changes(s, n) << " " << maxinvs(s, n) << " "
    << mininvs(s, n) << " " << suminvs(s, n) << " " << colvo101(s, n) << "
     " << prinvs(s, n) << " " << hashinvs(s, n) << endl; }
    }
    else {
        int n; cin >> n;
        string s1; cin >> s1;
        if (n <= 100) {
            string s2; cin >> s2;
            int finded = 0;
            for (int i = 0; i < n; i++) {
                int nar = 0;
                for (int j = 0; j < n; j++) {
                    if (s2[j] != s1[(i + j) % n]) { nar = 1; }
                }
                if (nar == 0) { finded = 1; }
            }
            if (finded == 1) {
                cout << "YES" << endl;
            } else { cout << "NO" << endl; }
        } else {
            long long r1, r2, r3, r4, r5, r6, r7, r8, r9, r10; cin >> r1
    >> r2 >> r3 >> r4 >> r5 >> r6 >> r7 >> r8 >> r9 >> r10;
            if (r1 == colvo1(s1, n) && r2 == maxposl1(s1, n) && r3 ==
    maxposl0(s1, n) && r4 == changes(s1, n) && r5 == maxinvs(s1, n) && r6
    == mininvs(s1, n) && r7 == suminvs(s1, n) && r8 == colvo101(s1, n) &&
    r9 == prinvs(s1, n) && r10 == hashinvs(s1, n)) {
                cout << "YES" << endl;
            } else { cout << "NO" << endl; }
        }
    }
    return 0;
}
```

Task D

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
int n;
void sapr(int a, int b) {
    cout << a << " " << b << endl;
    char s; cin >> s;
    if (s == '!') { exit(0); }
}
void swapper(int a, int b) {
    sapr(a, b);
    for (int i = 0; i < n - 1; i++) { sapr(1, 1); }
}
void rec(int v) {
    //        - 1,2,3,...,n, v -
    if (v < n) { rec(v + 1); }
    for (int i = v + 1; i <= n; i++) {
        swapper(v, i);
        rec(v + 1);
        swapper(v, i);
    }
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    cin >> n;
    rec(1);
    return 0;
}
```

Task E

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
```

```
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    /*
    vector <int> s = { 1,2,3,4,5 };
    for (int i = 0; i < 20; i++) {
        swap(s[0], s[3]);
        for (int j = 0; j < 4; j++) { s[j] = s[j + 1]; }
        s[4] = 15 - s[0] - s[1] - s[2] - s[3];
        for (int j = 0; j < 5; j++) { cout << s[j] << " "; }
        cout << "\n";
    }
    */
    int n, q; cin >> n >> q;
    vector <vector<int>> sosed(n);
    for (int i = 0; i < n; i++) {
        int k; cin >> k;
        sosed[i].resize(k);
        for (int j = 0; j < k; j++) { cin >> sosed[i][j]; }
    }
    vector <int> qs(q);
    for (int i = 0; i < q; i++) { cin >> qs[i]; }
    if (qs[q - 1] <= 100000) {
        vector <int> ism(n, 0);
        int pos = 0, start = 1;
        if (qs[0] == 0) { cout << 1 << " "; pos++; }
        for (int i = 1; i <= qs[q - 1]; i++) {
            int k = sosed[start - 1].size(), nextt = sosed[start - 1][ism
    [start - 1]% k];
            ism[start - 1]++; start = nextt;
            if (i == qs[pos]) { cout << start << " "; pos++; }
        }
    }
    return 0;
}
```

Task F

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
```

```
#include <set>
#include <map>
#include <queue>
#include <ctime>
#include <chrono>
#include <random>
#include <cmath>
using namespace std;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int n, m; cin >> n >> m;
    vector <string> s(n);
    for (int i = 0; i < n; i++) { cin >> s[i]; }
    vector <vector<int>> mas(n, vector<int>(m, 0));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            mas[i][j] = s[i][j] - '0';
        }
    }
    int ans = 0;
    vector <vector<int>> answer(n, vector<int>(m, 0));
    for (int i = 0; i < n / 2; i++) {
        for (int j = 0; j < m / 2; j++) {
            int res1 = mas[2 * i][2 * j] * mas[2 * i][2 * j + 1] + mas[2
   * i + 1][2 * j] * mas[2 * i + 1][2 * j + 1];
            int res2 = mas[2 * i][2 * j] * mas[2 * i + 1][2 * j] + mas[2
   * i][2 * j + 1] * mas[2 * i + 1][2 * j + 1];
            if (res1 > res2) {
                ans += res1;
                answer[2 * i][2 * j] = 3;
                answer[2 * i + 1][2 * j] = 3;
                answer[2 * i][2 * j + 1] = 4;
                answer[2 * i + 1][2 * j + 1] = 4;
            } else {
                ans += res2;
                answer[2 * i][2 * j] = 1;
                answer[2 * i + 1][2 * j] = 2;
                answer[2 * i][2 * j + 1] = 1;
                answer[2 * i + 1][2 * j + 1] = 2;
            }
        }
    }
    vector <vector<int>> answer2(n, vector<int>(m, 0));
    int ans2 = 0;
    for (int i = 0; i < n / 2; i++) {
        for (int j = 0; j < m / 2; j++) {
            int res1 = mas[2 * i + 1][2 * j + 1] * mas[2 * i + 1][(2 * j
   + 2) % m] + mas[(2 * i + 2) % n][2 * j + 1] * mas[(2 * i + 2) % n][(2
   * j + 2) % m];
            int res2 = mas[2 * i + 1][2 * j + 1] * mas[(2 * i + 2) % n][2
    * j + 1] + mas[2 * i + 1][(2 * j + 2) % m] * mas[(2 * i + 2) % n][(2
```

```
* j + 2) % m];
        if (res1 > res2) {
            ans2 += res1;
            answer2[2 * i + 1][2 * j + 1] = 3;
            answer2[(2 * i + 2)%n][2 * j + 1] = 3;
            answer2[2 * i + 1][(2 * j + 2) % m] = 4;
            answer2[(2 * i + 2)%n][(2 * j + 2)%m] = 4;
        }
        else {
            ans2 += res2;
            answer2[2 * i + 1][2 * j + 1] = 1;
            answer2[(2 * i + 2) % n][2 * j + 1] = 2;
            answer2[2 * i + 1][(2 * j + 2) % m] = 1;
            answer2[(2 * i + 2) % n][(2 * j + 2) % m] = 2;
        }
    }
}
if (ans2 > ans) {
    ans = ans2;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            answer[i][j] = answer2[i][j];
        }
    }
}
ans2 = 0;
for (int i = 0; i < n / 2; i++) {
    for (int j = 0; j < m / 2; j++) {
        int res1 = mas[2 * i + 1][2 * j] * mas[2 * i + 1][2 * j + 1]
+ mas[(2 * i + 2)%n][2 * j] * mas[(2 * i + 2)%n][2 * j + 1];
        int res2 = mas[2 * i + 1][2 * j] * mas[(2 * i + 2)%n][2 * j]
+ mas[2 * i + 1][2 * j + 1] * mas[(2 * i + 2)%n][2 * j + 1];
        if (res1 > res2) {
            ans2 += res1;
            answer2[2 * i + 1][2 * j] = 3;
            answer2[(2 * i + 2) % n][2 * j] = 3;
            answer2[2 * i + 1][2 * j + 1] = 4;
            answer2[(2 * i + 2) % n][2 * j + 1] = 4;
        }
        else {
            ans2 += res2;
            answer2[2 * i + 1][2 * j] = 1;
            answer2[(2 * i + 2)%n][2 * j] = 2;
            answer2[2 * i + 1][2 * j + 1] = 1;
            answer2[(2 * i + 2)%n][2 * j + 1] = 2;
        }
    }
}
if (ans2 > ans) {
    ans = ans2;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            answer[i][j] = answer2[i][j];
```

```
            }
        }
    }
    ans2 = 0;
    for (int i = 0; i < n / 2; i++) {
        for (int j = 0; j < m / 2; j++) {
            int res1 = mas[2 * i][2 * j + 1] * mas[2 * i][(2 * j + 2) % m
] + mas[2 * i + 1][2 * j + 1] * mas[2 * i + 1][(2 * j + 2) % m];
            int res2 = mas[2 * i][2 * j + 1] * mas[2 * i + 1][2 * j + 1]
+ mas[2 * i][(2 * j + 2) % m] * mas[2 * i + 1][(2 * j + 2) % m];
            if (res1 > res2) {
                ans2 += res1;
                answer2[2 * i][2 * j + 1] = 3;
                answer2[2 * i + 1][2 * j + 1] = 3;
                answer2[2 * i][(2 * j + 2) % m] = 4;
                answer2[2 * i + 1][(2 * j + 2) % m] = 4;
            }
            else {
                ans2 += res2;
                answer2[2 * i][2 * j + 1] = 1;
                answer2[2 * i + 1][2 * j + 1] = 2;
                answer2[2 * i][(2 * j + 2) % m] = 1;
                answer2[2 * i + 1][(2 * j + 2) % m] = 2;
            }
        }
    }
    if (ans2 > ans) {
        ans = ans2;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                answer[i][j] = answer2[i][j];
            }
        }
    }
    cout << ans << "\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (answer[i][j] == 1) {
                cout << "T";
            }
            else if (answer[i][j] == 2) {
                cout << "B";
            }
            else if (answer[i][j] == 3) {
                cout << "L";
            }
            else {
                cout << "R";
            }
        } cout << "\n";
    }
    return 0;
}
```