Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0330 – Никурадзе Дмитрий Андреевич

A	B	C	D	E	F	sum
100	100	100	100	20	67	487

Task A

```
//#define _CRT_SECURE_NO_DEPRECATE
//#pragma GCC optimize("O3")
//
// #pragma GCC optimize("Ofast")
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
#include <algorithm>
#include <map>
#include <deque>
#include <stdio.h>
#include <cmath>
#include <stdlib.h>
#include <bitset>
#include <string>

//#include <bits/stdc++.h>
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#define int long long
#define double long double
using ll = long long;
using namespace std;
```

```
const ll mod = 1e9 + 7;
const ll mod9 = 1e9 + 9;
const ll INF = 1e9 + 20;
const int sz_block = 40;
const ll SUPER_INF = 1e18 + 1;
const ll INF64 = 1e18;
#define pb push_back
#define get_time (double)clock() / (double)(CLOCKS_PER_SEC)
#ifndef ONLINE_JUDGE
#define debug(x) cerr << (#x) << ": " << x << endl;
#else
#define debug(x)
#endif
ll max(ll a, ll b) {
    return (a > b) ? a : b;
}
ll min(ll a, ll b) {
    return (a < b) ? a : b;
}
ll bs_sqrt(ll x) {
    ll left = 0, right = 2147483649;
    while (right > left) {
        ll mid = (left + right) / 2;
        if (mid * mid > x) {
            right = mid;
            continue;
        }
        left = mid + 1;
    }
    return left - 1;
}
int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        int c = a;
        a = b % a;
        b = c;
    }
    return a + b;
}
int bin_pow(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow(a, n / 2);
        return (t * t) % mod;
    } else {
        int t = bin_pow(a, n - 1);
        return (t * a) % mod;
    }
}
int bin_pow_no_mod(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow_no_mod(a, n / 2);
```

```
        return (t * t);
    } else {
        int t = bin_pow_no_mod(a, n - 1);
        return (t * a);
    }
}
vector<int> to_2(ll x) {
    vector<int> res;
    while (x > 0) {
        res.push_back(x % 2);
        x /= 2;
    }
    while (res.size() < sz_block) res.push_back(0);
    reverse(res.begin(), res.end());
    return res;
}
int mul(int a, int b, ll m) {
    return (ll(a) * b) % m;
}
int make_add(int a, int b) {
    if (a + b >= mod) return a + b - mod;
    return a + b;
}
int obr_to_10(vector<int> a) {
    int cur = 1, sum = 0;
    for (int i = a.size() - 1; i >= 0; i--) {
        sum += a[i] * cur;
        cur *= 2;
    }
    return sum;
}
int rev(int x) {
    return bin_pow(x, mod - 2);
}
int make_minus(int a, int b, int m) {
    if (a - b < 0) return a - b + m;
    return a - b;
}
int a, b, n, ans = 0;
int calc_to_n(int sum) {
    vector<int> dp(sum + 1, 0);
    dp[0] = 1;
    for (int i = 0; i < sum; i++) {
        int na = i + a;
        if (na > sum) continue;
        dp[na] += dp[i];
        int nb = i + b;
        if (nb > sum) continue;
        dp[nb] += dp[i];
    }
    return dp[sum];
}
void solve() {
```

```
    cin >> a >> b >> n;
    if (a > b) swap(a, b);
    if (n < a) {
        cout << 0;
        return;
    }
    int x = calc_to_n(n - a);
    ans = x * a;
    if (n >= b) {
        int y = calc_to_n(n - b);
        ans += b * y;
    }
    cout << ans << endl;
}
signed main() {
    //srand(time(NULL));
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    /*
    while (true) {
        debug(num_test);
        num_test++;
        gen();
    }
    */
    int t = 1;
    // cin >> t;
     //if (t != 2) return 0;
       // while (t--) get_cin();

    while (t--) {
        solve();
    }

    return 0;
}
/*
1
2 4 3
1 1 2
2 1 2
2 3 1
24fin0330
mersoamed
*/
```

Task B

```
//#pragma GCC optimize("O3")
```

```
//
// #pragma GCC optimize("Ofast")
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
#include <algorithm>
#include <map>
#include <deque>
#include <stdio.h>
#include <cmath>
#include <stdlib.h>
#include <bitset>
#include <string>

//#include <bits/stdc++.h>
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#define int long long
#define double long double
using ll = long long;
using namespace std;
const ll mod = 1e9 + 7;
const ll mod9 = 1e9 + 9;
const ll INF = 1e9 + 20;
const int sz_block = 40;
const ll SUPER_INF = 1e18 + 1;
const ll INF64 = 1e18;
#define pb push_back
#define get_time (double)clock() / (double)(CLOCKS_PER_SEC)
#ifndef ONLINE_JUDGE
#define debug(x) cerr << (#x) << ": " << x << endl;
#else
#define debug(x)
#endif
ll max(ll a, ll b) {
    return (a > b) ? a : b;
}
ll min(ll a, ll b) {
    return (a < b) ? a : b;
}
ll bs_sqrt(ll x) {
    ll left = 0, right = 2147483649;
    while (right > left) {
        ll mid = (left + right) / 2;
        if (mid * mid > x) {
            right = mid;
            continue;
```

```
        }
        left = mid + 1;
    }
    return left - 1;
}
int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        int c = a;
        a = b % a;
        b = c;
    }
    return a + b;
}
int bin_pow(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow(a, n / 2);
        return (t * t) % mod;
    } else {
        int t = bin_pow(a, n - 1);
        return (t * a) % mod;
    }
}
int bin_pow_no_mod(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow_no_mod(a, n / 2);
        return (t * t);
    } else {
        int t = bin_pow_no_mod(a, n - 1);
        return (t * a);
    }
}
int mul(int a, int b, ll m) {
    return (ll(a) * b) % m;
}
int make_add(int a, int b) {
    if (a + b >= mod) return a + b - mod;
    return a + b;
}
int rev(int x) {
    return bin_pow(x, mod - 2);
}
int make_minus(int a, int b, int m) {
    if (a - b < 0) return a - b + m;
    return a - b;
}
int abs(int a, int b) {
    if (a - b >= 0) return a - b;
    return b - a;
}
int n, x1, yy1, z1, x2, y2, z2;
int with_y() {
```

```
    int res = abs(yy1 - n) + abs(y2 + n);
    res = max(res, abs(x1 + n - x2));
    res = max(res, abs(z1 + n - z2));
    int ans = res;
    res = abs(yy1 + n) + abs(y2 - n);
    res = max(res, abs(x1 - n - x2));
    res = max(res, abs(z1 - n - z2));
    ans = min(ans, res);
    return ans;
}
int with_x() {
    int res = abs(x1 - n) + abs(x2 + n);
    res = max(res, abs(yy1 + n - y2));
    res = max(res, abs(z1 + n - z2));
    int ans = res;
    res = abs(x1 + n) + abs(x2 - n);
    res = max(res, abs(yy1 - n - y2));
    res = max(res, abs(z1 - n - z2));
    ans = min(ans, res);
    return ans;
}
int with_z() {
    int res = abs(z1 - n) + abs(z2 + n);
    res = max(res, abs(yy1 + n - y2));
    res = max(res, abs(x1 + n - x2));
    int ans = res;
    res = abs(z1 + n) + abs(z2 - n);
    res = max(res, abs(yy1 - n - y2));
    res = max(res, abs(x1 - n - x2));
    ans = min(ans, res);
    return ans;
}
void solve() {
    cin >> n >> x1 >> yy1 >> z1 >> x2 >> y2 >> z2;
    int res = max(abs(x1 - x2), abs(yy1 - y2));
    res = max(res, abs(z1 - z2));
    int res_y = with_y();
    int res_x = with_x();
    int res_z = with_z();
    int ans = min(res, res_x);
    ans = min(ans, res_y);
    ans = min(ans, res_z);
    cout << ans << endl;
}
signed main() {
    //srand(time(NULL));
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    /*
    while (true) {
        debug(num_test);
```

```
        num_test++;
        gen();
    }
    */
    int t = 1;
    // cin >> t;
     //if (t != 2) return 0;
       // while (t--) get_cin();

    while (t--) {
        solve();
    }

    return 0;
}
/*
1
2 4 3
1 1 2
2 1 2
2 3 1
24fin0330
mersoamed
*/
```

Task C

```
//#pragma GCC optimize("O3")
//
// #pragma GCC optimize("Ofast")
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
#include <algorithm>
#include <map>
#include <deque>
#include <stdio.h>
#include <cmath>
#include <stdlib.h>
#include <bitset>
#include <string>
#include <random>
//#include <bits/stdc++.h>
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
```

```
#define int long long
#define double long double
using ll = long long;
using namespace std;
const ll mod = 1e9 + 9;
const ll mod9 = 1e9 + 9;
const ll INF = 1e9 + 20;
const int sz_block = 40;
const int SZ = 15;
const ll SUPER_INF = 1e18 + 1;
const ll INF64 = 1e18;
#define pb push_back
#define get_time (double)clock() / (double)(CLOCKS_PER_SEC)
#ifndef ONLINE_JUDGE
#define debug(x) cerr << (#x) << ": " << x << endl;
#else
#define debug(x)
#endif
ll max(ll a, ll b) {
    return (a > b) ? a : b;
}
ll min(ll a, ll b) {
    return (a < b) ? a : b;
}
ll bs_sqrt(ll x) {
    ll left = 0, right = 2147483649;
    while (right > left) {
        ll mid = (left + right) / 2;
        if (mid * mid > x) {
            right = mid;
            continue;
        }
        left = mid + 1;
    }
    return left - 1;
}
int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        int c = a;
        a = b % a;
        b = c;
    }
    return a + b;
}
int bin_pow(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow(a, n / 2);
        return (t * t) % mod;
    } else {
        int t = bin_pow(a, n - 1);
        return (t * a) % mod;
    }
```

```
}
int bin_pow_no_mod(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow_no_mod(a, n / 2);
        return (t * t);
    } else {
        int t = bin_pow_no_mod(a, n - 1);
        return (t * a);
    }
}
int mul(int a, int b) {
    return (ll(a) * b) % mod;
}
int make_add(int a, int b) {
    if (a + b >= mod) return a + b - mod;
    return a + b;
}
int rev(int x) {
    return bin_pow(x, mod - 2);
}
int make_minus(int a, int b, int m) {
    if (a - b < 0) return a - b + m;
    return a - b;
}
int abs(int a, int b) {
    if (a - b >= 0) return a - b;
    return b - a;
}
int obr_to_10(vector<int> a) {
    int cur = 1, sum = 0;
    for (int i = a.size() - 1; i >= 0; i--) {
        sum += a[i] * cur;
        cur *= 10;
    }
    return sum;
}
const int base = 1e6 + 3;
int n;
int make_hash(vector<int> a) {
    int hash = 0;
    for (auto c : a) {
        hash = make_add(mul(hash, base), c);
    }
    return hash;
}
int lol(string s) {
    vector<int> a;
    deque<int> d;
    for (int i = 0; i < n; i++) {
        if (s[i] == '1') {
            d.push_back(1);
        } else {
```

```
            d.push_back(0);
        }
    }
    for (auto c : d) a.push_back(c);
    int hash = make_hash(a);
    return hash;
}
void incode() {
    string s;
    cin >> s;
    int hash = lol(s);
    cout << hash << endl;
}
void decode() {
    string s, from_incode;
    cin >> s;
    cin >> from_incode;
    vector<int> p;
    for (auto c : from_incode) p.push_back(c - 48);
    int must_hash = obr_to_10(p);
    int hash = lol(s);
    int poww = 1;
    for (int i = 0; i < n - 1; i++) poww = mul(poww, base);
    for (int i = 0; i <= n; i++) {
        if (must_hash == hash) {
            cout << "YES" << endl;
            return;
        }
        if (i == n) break;
        int fl = 0;
        if (s[i] == '1') fl = 1;
        hash = (hash - fl * poww + mod) % mod;
        hash = mul(hash, base);
        hash = (hash + fl) % mod;
    }
    cout << "NO" << endl;
}
void solve() {
    string type;
    cin >> type >> n;
    if (type == "first") {
        incode();
    } else {
        decode();
    }
}
signed main() {
    //srand(time(NULL));
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    /*
```

```
    while (true) {
        debug(num_test);
        num_test++;
        gen();
    }
    */
    int t = 1;
    // cin >> t;
     //if (t != 2) return 0;
       // while (t--) get_cin();

    while (t--) {
        solve();
    }

    return 0;
}
/*
second
6
110010
215434162
*/
```

Task D

```
2 6
317233
229870
```

Task E

```
.6
```

Task F

```
//#pragma GCC optimize("O3")
//
// #pragma GCC optimize("Ofast")
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <unordered_map>
#include <iomanip>
```

```
#include <algorithm>
#include <map>
#include <deque>
#include <stdio.h>
#include <cmath>
#include <stdlib.h>
#include <bitset>
#include <string>
#include <random>
//#include <bits/stdc++.h>
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#define int long long
#define double long double
using ll = long long;
using namespace std;
const ll mod = 1e9 + 9;
const ll mod9 = 1e9 + 9;
const ll INF = 1e9 + 20;
const int sz_block = 40;
const int SZ = 15;
const ll SUPER_INF = 1e18 + 1;
const ll INF64 = 1e18;
#define pb push_back
#define get_time (double)clock() / (double)(CLOCKS_PER_SEC)
#ifndef ONLINE_JUDGE
#define debug(x) cerr << (#x) << ": " << x << endl;
#else
#define debug(x)
#endif
ll max(ll a, ll b) {
    return (a > b) ? a : b;
}
ll min(ll a, ll b) {
    return (a < b) ? a : b;
}
ll bs_sqrt(ll x) {
    ll left = 0, right = 2147483649;
    while (right > left) {
        ll mid = (left + right) / 2;
        if (mid * mid > x) {
            right = mid;
            continue;
        }
        left = mid + 1;
    }
    return left - 1;
}
int gcd(int a, int b) {
    while (a > 0 && b > 0) {
        int c = a;
        a = b % a;
        b = c;
    }
```

```
    return a + b;
}
int bin_pow(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow(a, n / 2);
        return (t * t) % mod;
    } else {
        int t = bin_pow(a, n - 1);
        return (t * a) % mod;
    }
}
int bin_pow_no_mod(int a, int n) {
    if (n == 0) return 1;
    if (n % 2 == 0) {
        int t = bin_pow_no_mod(a, n / 2);
        return (t * t);
    } else {
        int t = bin_pow_no_mod(a, n - 1);
        return (t * a);
    }
}
int mul(int a, int b) {
    return (ll(a) * b) % mod;
}
int make_add(int a, int b) {
    if (a + b >= mod) return a + b - mod;
    return a + b;
}
int rev(int x) {
    return bin_pow(x, mod - 2);
}
int make_minus(int a, int b, int m) {
    if (a - b < 0) return a - b + m;
    return a - b;
}
int abs(int a, int b) {
    if (a - b >= 0) return a - b;
    return b - a;
}
int obr_to_10(vector<int> a) {
    int cur = 1, sum = 0;
    for (int i = a.size() - 1; i >= 0; i--) {
        sum += a[i] * cur;
        cur *= 10;
    }
    return sum;
}
int n, m, best = 0;
vector<vector<int> > a;
vector<vector<char> > res, out;
void make_very_good() {
    int ans = 0;
```

```
    res = vector<vector<char> >(n, vector<char>(m, 0));
    for (int i = 0; i < n; i += 2) {
        for (int j = 0; j < m; j += 2) {
            int f = a[i][j] * a[i][j + 1] + a[i + 1][j] * a[i + 1][j +
    1];
            int s = a[i][j] * a[i + 1][j] + a[i][j + 1] * a[i + 1][j +
    1];
            if (f >= s) {
                res[i][j] = 'L';
                res[i][j + 1] = 'R';
                res[i + 1][j] = 'L';
                res[i + 1][j + 1] = 'R';
            } else {
                res[i][j] = 'T';
                res[i + 1][j] = 'B';
                res[i][j + 1] = 'T';
                res[i + 1][j + 1] = 'B';
            }
            ans += max(f, s);
        }
    }
    if (ans > best) {
        best = ans;
        out = res;
    }
}
void calc() {
    int may = 0;
    res = vector<vector<char> >(n, vector<char>(m, '.'));
    vector<vector<int> > cur(n, vector<int>(m, 0));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (cur[i][j]) continue;
            int x = 0, niz = -INF, r = -INF;
            if (i != n - 1 && cur[i + 1][j] == 0) {
                niz = a[i][j] * a[i + 1][j];
            }
            if (j != m - 1 && cur[i][j + 1] == 0) {
                r = a[i][j] * a[i][j + 1];
            }
            may += max(r, niz);
            if (niz > r) {
                res[i][j] = 'T';
                res[i + 1][j] = 'B';
                cur[i][j] = 1;
                cur[i + 1][j] = 1;
            } else {
                res[i][j] = 'L';
                res[i][j + 1] = 'R';
                cur[i][j] = 1;
                cur[i][j + 1] = 1;
            }
        }
```

```
    }
    if (may > best) {
        best = may;
        out = res;
    }
}
void with_rand() {
    int may = 0;
    res = vector<vector<char> >(n, vector<char>(m, '.'));
    vector<vector<int> > cur(n, vector<int>(m, 0));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (cur[i][j]) continue;
            int x = 0, niz = -INF, r = -INF;
            if (i != n - 1 && cur[i + 1][j] == 0) {
                niz = a[i][j] * a[i + 1][j];
            }
            if (j != m - 1 && cur[i][j + 1] == 0) {
                r = a[i][j] * a[i][j + 1];
            }
            x = rand() % 2;
            if (x == 0 && niz != -INF) {
                may += niz;
                res[i][j] = 'T';
                res[i + 1][j] = 'B';
                cur[i][j] = 1;
                cur[i + 1][j] = 1;
            } else if (x == 0 && niz == -INF) {
                may += r;
                res[i][j] = 'L';
                res[i][j + 1] = 'R';
                cur[i][j] = 1;
                cur[i][j + 1] = 1;
            }
            if (x == 1 && r != -INF) {
                may += r;
                res[i][j] = 'L';
                res[i][j + 1] = 'R';
                cur[i][j] = 1;
                cur[i][j + 1] = 1;
            } else if (x == 1 && r == -INF) {
                may += niz;
                res[i][j] = 'T';
                res[i + 1][j] = 'B';
                cur[i][j] = 1;
                cur[i + 1][j] = 1;
            }
        }
    }
    if (may > best) {
        best = may;
        out = res;
    }
```

```
}
void solve() {
    cin >> n >> m;
    res = vector<vector<char> >(n, vector<char>(m, '.'));
    a = vector<vector<int> >(n, vector<int>(m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            char c; cin >> c;
            int x = c - 48;
            a[i][j] = x;
        }
    }
    make_very_good();
    calc();
    for (int i = 0; i < 1000; i++) {
        with_rand();
    }
    cout << best << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << out[i][j];
        }
        cout << endl;
    }
    cout << endl;
}
signed main() {
    srand(time(NULL));
    cout << fixed << setprecision(20);
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    /*
    while (true) {
        debug(num_test);
        num_test++;
        gen();
    }
    */
    int t = 1;
    // cin >> t;
     //if (t != 2) return 0;
       // while (t--) get_cin();

    while (t--) {
        solve();
    }

    return 0;
}
/*
second
6
```

```
110010
215434162
*/
```