Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0129 – Данилов Андрей Михайлович

A	B	C	D	E	F	sum
100	50	100	35	20	89	394

Task A

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;



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

    long long a, b, n;
    cin >> a >> b >> n;
    map<long long , long long > was;
```

```
    long long  ans = 0;
    for (long long  j = 0; j < n; j++) {
        vector<long long > dp(2*n + 1);
        dp[j] = 1;
        for (long long  i = j; i < n+j; i++) {
            if (dp[i] == 0) continue;
            if (i + a <= 2*n && was[(i+a)%n]==0) {
                dp[i + a] += dp[i];
            }
            if (i + b <= 2*n && was[(i+b)%n]==0) {
                dp[i + b] += dp[i];
            }
        }
        was[j] = 1;
        for (long long  i = 0; i <= n; i++) {
            //cout << dp[i] << " ";
        }
        //cout << "\n";
        ans += dp[j+n];
        //cout << j << " " << ans << "\n";
    }
    cout << ans << "\n";
}
```

Task B

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;

long long n;


struct dot {
    long long x, y, z;

    dot(long long a, long long b, long long c) {
        x = a;
```

```
        y = b;
        z = c;
    }

    dot () {
        x = 0;
        y = 0;
        z = 0;
    }
};

dot rev(dot a) {
    return {-a.x, -a.y, -a.z};
}

long long rast(dot a, dot b) {
    if (a.x+a.y+a.z!=0 || b.x+b.y+b.z!=0) return 1e12;
    if (a.x>n || a.y>n ||  a.z>n ||  b.x>n ||  b.y>n ||  b.z>n) return 1
   e12;
    return (abs(a.x - b.x) + abs(a.y - b.y) + abs(a.z - b.z))/2;
}


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

    cin >> n;
    long long x1, y1, z1, x2, y2, z2;
    cin >> x1 >> y1 >> z1;
    cin >> x2 >> y2 >> z2;
    dot f1 = {x1, y1, z1};
    dot f2 = {x2, y2, z2};
    long long ans = 1e18;
    long long cur1, cur2, cur3, cur4;
    cur1 = ans; cur2 = 0; cur3 = 0; cur4 = 0;

    cur1 = rast(f1, f2);
    ans = min(ans, cur1);
    if (n<=12) {
        for (long long i = -n; i <= n; i++) {
            for (long long y = -n; y <= n; y++) {
                for (long long ie = -n; ie <= n; ie++) {
                    for (long long ye = -n; ye <= n; ye++) {
                        long long z = -(i + y);
                        long long ze = -(ie + ye);
                        if (abs(i) != n && abs(y) != n && abs(z) != n)
   continue;
                        if (abs(ie) != n && abs(ye) != n && abs(ze) != n)
    continue;
                        dot f3 = {i, y, z};
                        dot f4 = rev(f3);
```

```
                    dot f5 = {ie, ye, ze};
                    dot f6 = rev(f5);
                    if (abs(z) > n) continue;
                    if (abs(ze) > n) continue;
                    ans = min(ans, (rast(f1, f3) + rast(f2, f3) ) );
                    ans = min(ans, (rast(f1, f3) + rast(f2, f4) ));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f4) + rast(f2, f3) ));
                    ans = min(ans, (rast(f1, f4) + rast(f2, f4) ));


                    ans = min(ans, (rast(f1, f3) + rast(f2, f5) ) +
rast(f5, f3)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f5) +
rast(f4, f5)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f5) + rast(f2, f3) +
rast(f5, f3)));
                    ans = min(ans, (rast(f1, f5) + rast(f2, f4) +
rast(f5, f4)));


                    ans = min(ans, (rast(f1, f3) + rast(f2, f6) ) +
rast(f6, f3)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f6) +
rast(f4, f6)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f6) + rast(f2, f3) +
rast(f6, f3)));
                    ans = min(ans, (rast(f1, f6) + rast(f2, f4) +
rast(f6, f4)));


                    ans = min(ans, (rast(f1, f3) + rast(f2, f5) ) +
rast(f6, f3)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f5) +
rast(f4, f6)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f5) + rast(f2, f3) +
rast(f6, f3)));
                    ans = min(ans, (rast(f1, f5) + rast(f2, f4) +
rast(f6, f4)));


                    ans = min(ans, (rast(f1, f3) + rast(f2, f6) ) +
rast(f5, f3)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f6) +
rast(f4, f5)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
```

```
"\n";
                    ans = min(ans, (rast(f1, f6) + rast(f2, f3) +
rast(f5, f3)));
                    ans = min(ans, (rast(f1, f6) + rast(f2, f4) +
rast(f5, f4)));



                    ans = min(ans, (rast(f1, f3) + rast(f2, f5) ) +
rast(f5, f4)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f5) +
rast(f4, f5)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f5) + rast(f2, f3) +
rast(f5, f4)));
                    ans = min(ans, (rast(f1, f5) + rast(f2, f4) +
rast(f5, f3)));

                    ans = min(ans, (rast(f1, f3) + rast(f2, f6) ) +
rast(f6, f4)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f6) +
rast(f3, f6)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f6) + rast(f2, f3) +
rast(f6, f4)));
                    ans = min(ans, (rast(f1, f6) + rast(f2, f4) +
rast(f6, f3)));



                    ans = min(ans, (rast(f1, f3) + rast(f2, f5) ) +
rast(f5, f4)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f5) +
rast(f4, f5)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f5) + rast(f2, f3) +
rast(f5, f4)));
                    ans = min(ans, (rast(f1, f5) + rast(f2, f4) +
rast(f5, f3)));

                    ans = min(ans, (rast(f1, f3) + rast(f2, f6) ) +
rast(f6, f4)  );
                    ans = min(ans, (rast(f1, f4) + rast(f2, f6) +
```

```
rast(f3, f6)));
                    //cout << rast(f1, f3) << " " << rast(f2, f4) <<
"\n";
                    ans = min(ans, (rast(f1, f6) + rast(f2, f3) +
rast(f6, f4)));
                    ans = min(ans, (rast(f1, f6) + rast(f2, f4) +
rast(f6, f3)));
                }
            }

        }
    }



    for (int i1  = -n; i1 <= n; i1+=n) {
        for (int i2  = -n; i2 <= n; i2+=n) {
            for (int j1  = -n; j1 <= n; j1++) {
                for (int j2  = -n; j2 <= n; j2++) {
                    for (int z1  = -n; z1 <= n; z1++) {
                        for (int z2  = -n; z2 <= n; z2++) {
                            if (i1+j1+z1 != 0 || i2+j2+z2!=0)
continue;
                            dot e1 = {i1, j1, z1};
                            dot e2 = {i2, j2, z2};
                            ans = min(ans, rast(f1, e1)+rast(e1, e2)+
rast(e2, f2));
                            dot e3 =rev(e1);
                            dot e4 = rev(e2);
                            ans = min(ans, rast(f1, e1)+rast(e1, e4)+
rast(e2, f2));
                            ans = min(ans, rast(f1, e1)+rast(e3, e2)+
rast(e2, f2));
                            ans = min(ans, rast(f1, e1)+rast(e3, e3)+
rast(e2, f2));
                            ans = min(ans, rast(f1, e1)+rast(e1, e2)+
rast(e2, f2));
                            ans = min(ans, rast(f1, e1)+rast(e1, e4)+
rast(e4, f2));
                            ans = min(ans, rast(f1, e1)+rast(e3, e2)+
rast(e4, f2));
                            ans = min(ans, rast(f1, e1)+rast(e3, e3)+
rast(e4, f2));
                            ans = min(ans, rast(f1, e1)+rast(e1, e2)+
rast(e4, f2));
                        }
                    }
                }
            }
        }
    }
}
```

```
    cout << ans << "\n";
}
```

Task C

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;

long long mod = 1e15;

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

    string t;
    cin >> t;
    long long e1 = 3;
    long long e2 = 19;
    long long e3 = 13;
    vector<long long> p1(1e6+10);
    p1[0] = 1;
    for (long long i = 1; i< 1e6+5; i++) {
        p1[i] = ((p1[i-1]*e1)%mod+mod)%mod;
    }
    vector<long long> p2(1e6+10);
    p2[0] = 1;
    for (long long i = 1; i< 1e6+5; i++) {
        p2[i] = ((p2[i-1]*e2)%mod+mod)%mod;
    }
    vector<long long> p3(1e6+10);
    p3[0] = 1;
    for (long long i = 1; i< 1e6+5; i++) {
        p3[i] = ((p3[i-1]*e3)%mod+mod)%mod;
    }
```

```
if (t == "first") {
    long long n;
    cin >> n;
    string s;
    cin >> s;
    long long hash1 = 0;
    for (long long i = 0; i < n; i++) {
        if (s[i]=='1') {
            hash1 = ((hash1+2*p1[i])%mod+mod)%mod;
        } else {
            hash1 = ((hash1+p1[i])%mod+mod)%mod;
        }
    }
    long long hash2 = 0;
    for (long long i = 0; i < n; i++) {
        if (s[i]=='1') {
            hash2 = ((hash2+2*p2[i])%mod+mod)%mod;
        } else {
            hash2 = ((hash2+p2[i])%mod+mod)%mod;
        }
    }

    long long hash3 = 0;
    for (long long i = 0; i < n; i++) {
        if (s[i]=='1') {
            hash3 = ((hash3+2*p3[i])%mod+mod)%mod;
        } else {
            hash3 = ((hash3+p3[i])%mod+mod)%mod;
        }
    }

    cout << hash1 << "!" << hash2 << "!" << hash3<< "!" << n <<"\n";

} else {
    long long n;
    cin >> n;
    string s2;
    string s1;
    cin >> s1 >> s2;
    s1 = s1+s1;
    long long hash1 = 0;
    long long hash2 = 0;
    long long hash3 = 0;
    long long h1, h2, h3, h4;

    h1 = 0; h2  =0; h3 = 0, h4 = 0;
    long long u = 0;
    long long ff = 1;
    while (u<s2.size()) {
        if (s2[u]=='!') {
            ff++;
        } else {
            int e = s2[u]-'0';
```

```
        if (ff==1) {
            h1 = h1*10+e;
        }
        if (ff==2) {
            h2 = h2*10+e;
        }
        if (ff==3) {
            h3 = h3*10+e;
        }
        if (ff==4) {
            h4 = h4*10+e;
        }
    }
    u++;
}



for (long long i = 0; i < n; i++) {
    if (s1[i]=='1') {
        hash1 = ((hash1+2*p1[i])%mod+mod)%mod;
    } else {
        hash1 = ((hash1+p1[i])%mod+mod)%mod;
    }
}
for (long long i = 0; i < n; i++) {
    if (s1[i]=='1') {
        hash2 = ((hash2+2*p2[i])%mod+mod)%mod;
    } else {
        hash2 = ((hash2+p2[i])%mod+mod)%mod;
    }
}
for (long long i = 0; i < n; i++) {
    if (s1[i]=='1') {
        hash3 = ((hash3+2*p3[i])%mod+mod)%mod;
    } else {
        hash3 = ((hash3+p3[i])%mod+mod)%mod;
    }
}
long long fl = 0;
for (long long i = 0; i < n; i++) {
    //cout << hash1 << " " << h1 << "\n";
    //cout << hash2 << " " << h2 << "\n";
    //cout << hash3 << " " << h3 << "\n";
    if (hash1 == h1 && hash2 == h2 && hash3 == h3  && n==h4) {
        fl = 1;
    }
    if (s1[i]=='1') {
        hash1 = ((hash1-2*p1[i])%mod+mod)%mod;
        hash1 = ((hash1+2*p1[i+n])%mod+mod)%mod;
    } else {
        hash1 = ((hash1-p1[i])%mod+mod)%mod;
        hash1 = ((hash1+p1[i+n])%mod+mod)%mod;
```

```
            }
            h1 = ((h1*e1)%mod+mod)%mod;
            if (s1[i]=='1') {
                hash2 = ((hash2-2*p2[i])%mod+mod)%mod;
                hash2 = ((hash2+2*p2[i+n])%mod+mod)%mod;
            } else {
                hash2 = ((hash2-p2[i])%mod+mod)%mod;
                hash2 = ((hash2+p2[i+n])%mod+mod)%mod;
            }
            h2 = ((h2*e2)%mod+mod)%mod;
            if (s1[i]=='1') {
                hash3 = ((hash3-2*p3[i])%mod+mod)%mod;
                hash3 = ((hash3+2*p3[i+n])%mod+mod)%mod;
            } else {
                hash3 = ((hash3-p3[i])%mod+mod)%mod;
                hash3 = ((hash3+p3[i+n])%mod+mod)%mod;
            }
            h3 = ((h3*e3)%mod+mod)%mod;
        }
        if (hash1 == h1 && hash2 == h2 && hash3 == h3 && n==h4) {
            fl = 1;
        }
        if (fl) {
            cout << "YES\n";
        } else {
            cout << "NO\n";
        }
    }

}
//619!5090139!773709!6
```

Task D

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;
```

```
int main() {
    //ios::sync_with_stdio(false);
    //cin.tie(nullptr);
    //cout.tie(nullptr);

    int n;
    cin >> n;
    int x = 100;
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "1 2" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "1 3" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "2 3" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
```

```
        }
    }
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "2 4" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "3 4" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
    for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "1 4" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
```

```
                exit(0);
            }
        }
        cout << "1 5" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "2 5" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "3 5" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
    }
for (int j = 0; j < x; j++) {
        char q;
        for (int i = 0; i <= n; i++) {
            cout << "1 1" << endl;
            cin >> q;
            if (q == '!') {
                exit(0);
            }
        }
        cout << "4 5" << endl;
        cin >> q;
        if (q == '!') {
            exit(0);
        }
```

```
    }
}
```

Task E

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;



vector<pair<long long, vector<long long>>> g;
map<long long, long long> ans;
long long curt = 0;
long long v = 0;
map<long long, long long> was;

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

    int n, q;
    cin >> n >> q;
    g.resize(n);
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        g[i].first = 0;
        g[i].second.resize(x);
        for (int j = 0; j < x; j++) {
            cin >> g[i].second[j];
            g[i].second[j]--;
        }
    }
    vector<long long> t(q);
    for (int i = 0; i < q; i++) {
        cin >> t[i];
```

```
    }
    long long mod = -1;

    while (curt<100002) {
        //cout << v << " " << curt << endl;
        ans[curt] = v;
        int h = g[v].first;
        g[v].first = (g[v].first+1)%(g[v].second.size());
        v = g[v].second[h];
        curt++;
    }
    for (int i = 0; i < q; i++) {
        cout << ans[t[i]] + 1 << " ";
    }
}
```

Task F

```
#include <iostream>
#include <vector>
#include <queue>
#include <algorithm>
#include <map>
#include <string>
#include <cmath>
#include <set>
#include <deque>
#include <stack>
#include <iomanip>
#include <cmath>
#include <cassert>
#include <bitset>
#include <unordered_map>
//#pragma GCC optimize("O3,unroll-loops")
//#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
using namespace std;





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

    int h, w;
    cin >> h >> w;
    long long ans = 0;
    vector<vector<int>> p(h, vector<int> (w, 0));
    vector<vector<char>> pole(h, vector<char> (w, '.'));
    set<pair<int, pair<int, int>>> s;
```

```
for (int i= 0; i < h; i++) {
    for (int j = 0; j < w; j++) {
        char e;
        cin >> e;
        p[i][j] = e-'0';
        pair<int, int> d = {i, j};
        s.insert({p[i][j], d});
    }
}
vector<pair<int, int>> a;
a.push_back({0, 1});
a.push_back({0, -1});
a.push_back({1, 0});
a.push_back({-1, 0});
int cnt = 0;
while (!s.empty() && cnt/3<h*w) {
    /*for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << pole[i][j];
        }
        cout << "\n";
    }*/
    cnt++;
    auto t = s.end();
    t--;
    auto u = *t;
    s.erase(u);
    vector<pair<int, pair<int, int>>> rr;
    auto y = u.second;
    for (auto q : a) {
        pair<int, int> k = {y.first+q.first, y.second+q.second};
        if (k.first>=0 && k.first<h && k.second>=0 && k.second<w &&
pole[k.first][k.second] == '.') {
            rr.push_back({p[k.first][k.second], k});
        }
    }
    std::sort(rr.begin(), rr.end());
    if (rr.empty()) continue;
    ans += u.first*rr.back().first;
    //cout << y.first << " " << y.second << "\n";
    //cout <<rr.back().second.first << " " << rr.back().second.second
 << "\n";
    //cout << "\n";
    if (rr.back().second.first>y.first) {
        pole[y.first][y.second] = 'T';
        pole[rr.back().second.first][rr.back().second.second] = 'B';
    }
    if (rr.back().second.first<y.first) {
        pole[y.first][y.second] = 'B';
        pole[rr.back().second.first][rr.back().second.second] = 'T';
    }
    if (rr.back().second.second>y.second) {
        pole[y.first][y.second] = 'L';
```

```
            pole[rr.back().second.first][rr.back().second.second] = 'R';
        }
        if (rr.back().second.second<y.second) {
            pole[y.first][y.second] = 'R';
            pole[rr.back().second.first][rr.back().second.second] = 'L';
        }
        s.erase(rr.back());
    }

    cout << ans << "\n";
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            cout << pole[i][j];
        }
        cout << "\n";
    }
}
```