

Задача А:

```
/**
 *   author:      fractal
 **/

#include <bits/stdc++.h>
using namespace std;

#define F first
#define S second
#define mp make_pair
#define pb push_back
#define pf push_front
#define ppb pop_back
#define ppf pop_front
#define speed ios_base::sync_with_stdio(0),cin.tie(0),cout.tie(0)
#define sz(x) (int)x.size()
#define len(x) (int)strlen(x)
#define all(x) x.begin(), x.end()
#define debug cerr << "OK\n";
#define ub upper_bound
#define lb lower_bound
#define make_unique(x) sort(all(x)), x.erase(unique(all(x)), x.end())

mt19937 bruh(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rofl(chrono::steady_clock::now().time_since_epoch().count());

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<pii> vpii;
typedef vector<pll> vpll;
typedef set<int> si;
typedef set<ll> sll;
typedef set<pii> spii;
```

```

typedef set<pll> spll;
typedef multiset <int> msi;
typedef multiset <ll> msl;
typedef map <int, int> mi;
typedef map <ll, ll> mll;

const int N = 2e5 + 2;
const int M = 1e5;
const int mod = 0;
const int inf = 2e9 + 3;
const ll INF = 1e15;
const ld pi2 = 2.0 * 3.141592653589793;
const ld pi = 3.141592653589793;
const ld eps = 1e-12;

const int dx[4] = {1, -1, 0, 0};
const int dy[4] = {0, 0, -1, 1};

#ifdef PC
    #define gcd __gcd
#endif

void files(string s = "main") {
    #ifdef PC
        freopen((s + ".in").c_str(), "r", stdin);
        freopen((s + ".out").c_str(), "w", stdout);
    #endif
}

int add(int a, int b) {
    if (a + b < 0) return a + b + mod;
    if (a + b >= mod) return a + b - mod;
    return a + b;
}

int mul(int a, int b) {
    return a * 1LL * b % mod;
}

int binpow(int a, int n) {

```

```

int ret = 1;
while (n) {
    if (n & 1) ret = mul(ret, a);
    a = mul(a, a);
    n >>= 1;
}
return ret;
}

```

```

int n;
ll t[N], x[N];
vector<int> q[N];
priority_queue<int, vector<int>, greater<int>> s;

```

```

bool check(ll mx) {
    for (int i = 1; i <= n; ++i) {
        q[i].clear();
    }
    while (sz(s)) s.pop();
    for (int i = 1; i <= n; ++i) {
        //|b[i] - p[i]| <= mx / a[i]
        ll rad = inf;
        if (t[i]) rad = mx / t[i];
        int l = max(1ll, x[i] - rad), r = min(n * 1ll, x[i] + rad);
        q[l].pb(r);
    }
    for (int i = 1; i <= n; ++i) {
        for (auto j : q[i])
            s.push(j);
        if (sz(s) && s.top() < i) return 0;
        if (s.empty()) return 0;
        s.pop();
    }
    return 1;
}

```

```

int main() {
    speed;
    cin >> n;
    for (int i = 1; i <= n; ++i) {

```

```
    cin >> x[i] >> t[i];
}
ll tl = 0, tr = INF, ans = -1;

while (tl <= tr) {
    ll tm = tl + tr >> 1ll;
    if (check(tm))
        tr = tm - 1, ans = tm;
    else
        tl = tm + 1;
}
cout << ans << '\n';
}
```

Задача В:

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

#define all(x) (x).begin(), (x).end()
#define sz(x) (int)(x).size()

using namespace std;

typedef long long ll;

struct Line {
    ll k, b;

    Line() : k(0), b(0) {}

    Line(ll _k, ll _b) : k(_k), b(_b) {}

    ll get(ll x) {
        return k * x + b;
    }

    double inter(Line &o) {
        return (double) (b - o.b) / (o.k - k);
    }
};

bool good(Line a, Line b, Line c) {
    return a.inter(b) < b.inter(c);
}

struct CHT : public vector<Line> {
    int ptr;
    function<bool(ll, ll)> cmp;

    CHT() : ptr(0), cmp(less<ll>()) {}

    CHT(function<bool(ll, ll)> _cmp) : ptr(0), cmp(_cmp) {}

    void add(Line l) {
```

```

if (!empty() && l.k == back().k) {
    if (cmp(l.b, back().b)) {
        pop_back();
    }
    else {
        return;
    }
}

while (size() >= 2 && !good(end()[-2], end()[-1], l)) {
    pop_back();
}

push_back(l);
}

ll get_ptr(ll x) {
    while (ptr + 1 < (int) size() && cmp(at(ptr + 1).get(x), at(ptr).get(x))) {
        ptr++;
    }

    ptr = min(ptr, (int) size() - 1);
    return at(ptr).get(x);
}

ll get(ll x) {
    int l = 0, r = (int) size() - 1;

    while (l != r) {
        int m = (l + r) >> 1;

        if (cmp(at(m).get(x), at(m + 1).get(x))) {
            r = m;
        }
        else {
            l = m + 1;
        }
    }

    return at(l).get(x);
}

```

```

    }
};

const int MAXN = (int) 5e5 + 5;
const ll LINF = (ll) 1e18;

ll L[MAXN], R[MAXN], H[MAXN];
ll C[MAXN], prefC[MAXN];
ll gap[MAXN], prefGap[MAXN];
ll dif[MAXN], prefDif[MAXN];
ll dot[MAXN], prefDot[MAXN];
ll dpM[MAXN], dpR[MAXN];
int n, m;

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

    cin >> n >> m;

    for (int i = 1; i <= n; i++) {
        cin >> L[i] >> R[i] >> H[i] >> C[i];
    }

    for (int i = 1; i <= n - 1; i++) {
        gap[i] = L[i + 1] - R[i];
        dif[i] = abs(H[i] - H[i + 1]);
    }

    for (int i = 1; i <= n; i++) {
        prefDif[i] = prefDif[i - 1] + dif[i];
        prefGap[i] = prefGap[i - 1] + gap[i];
        prefC[i] = prefC[i - 1] + C[i];
    }

    for (int i = 1; i <= n - 1; i++) {
        dot[i] = gap[i] * prefC[i];
        prefDot[i] = prefDot[i - 1] + dot[i];
    }
}

```

```
dpM[0] = LINF;  
dpR[0] = 0;
```

```
CHT hM, hR;
```

```
for (int i = 1; i <= n; i++) {  
    Line l;
```

```
    l.k = -prefC[i - 1];  
    l.b = dpR[i - 1] + H[i] - prefDif[i - 1] + prefGap[i - 1] * prefC[i - 1] - prefDot[i - 1];  
    hM.add(l);
```

```
    dpM[i] = hM.get_ptr(prefGap[i - 1]);  
    dpM[i] += (prefDot[i - 1]);
```

```
    l.k = -prefGap[i - 1];  
    l.b = prefDot[i - 1] + dpM[i];  
    hR.add(l);
```

```
    dpR[i] = hR.get_ptr(prefC[i]);  
    dpR[i] += (H[i] + prefDif[i - 1] - prefDot[i - 1] + prefGap[i - 1] * prefC[i]);
```

```
}
```

```
cout << dpR[n] + m << '\n';  
return 0;
```

```
}
```


Задача С:

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

#define f first
#define s second
#define pb push_back
#define mp make_pair

using namespace std;

typedef long long ll;
typedef pair<int, int> pii;
typedef pair<long long, long long> pll;

const int N = 300500, inf = 1e9, mod = 998244353;
const ll INF = 1e18;

struct Sqrt_Set {
    bool was[N];
    int ptr;

    void init() {
        for (int i = 0; i < N; i++)
            was[i] = 1;
        ptr = 0;
    }

    void erase(int x) {
        was[x] = 0;
    }

    int get() {
        while (!was[ptr])
            ptr++;
        return ptr;
    }
} ST;
```

```
int n, q;  
int a[N], b[N];  
int ql[N], qr[N];  
const int K = 600;
```

```
int ans[N];
```

```
int qpp[N];  
int qp[N], qpn;
```

```
int p[N], pval[N];
```

```
int pp(int v) {  
    return p[v] == v ? v : p[v] = pp(p[v]);  
}
```

```
int m, ca[N], cb[N];  
int cx_n, cx[N], cp[N];  
int c_timer, c_was[N];
```

```
bool in_block[N];
```

```
vector<int> g[N];  
int was[N], timer;  
int V, E, MX;
```

```
void dfs(int v) {  
    V++;  
    E += g[v].size();  
    MX = max(MX, pval[v]);  
    was[v] = timer;  
    for (auto to: g[v])  
        if (was[to] != timer)  
            dfs(to);  
}
```

```

int get(int l, int r) {
    timer++;

    int mex = n;
    m = 0;

    for (int i = l; i <= r; i++) {
        ca[m] = pp(a[i]);
        cb[m] = pp(b[i]);
        m++;
    }

    mex = min(mex, ST.get());

    for (int i = 0; i < m; i++) {
        int v = ca[i];
        int u = cb[i];
        g[v].emplace_back(u);
        g[u].emplace_back(v);
    }
    for (int i = 0; i < cx_n; i++) {
        int x = cx[i];
        if (was[x] == timer || pval[x] >= mex)
            continue;
        V = E = MX = 0;
        dfs(x);
        if (E / 2 + 1 == V)
            mex = min(mex, MX);
    }
    for (int i = 0; i < m; i++) {
        int v = ca[i];
        int u = cb[i];
        g[v].clear();
        g[u].clear();
    }

    return mex;
}

```

```

void solve(int L, int R) {

```

```
ST.init();
for (int i = 0; i <= n + 1; i++) {
    p[i] = i;
    pval[i] = i;
    in_block[i] = 0;
}
```

```
for (int i = L; i <= R; i++) {
    int v = a[i];
    int u = b[i];
    in_block[v] = in_block[u] = 1;
    ST.erase(v);
    ST.erase(u);
}
```

```
c_timer++;
cx_n = 0;
for (int i = L; i <= R; i++) {
    {
        int x = a[i];
        if (c_was[x] != c_timer) {
            cx[cx_n++] = x;
            c_was[x] = c_timer;
        }
    }
    {
        int x = b[i];
        if (c_was[x] != c_timer) {
            cx[cx_n++] = x;
            c_was[x] = c_timer;
        }
    }
}
for (int i = 0; i < cx_n; i++)
    cp[cx[i]] = i;
```

```
qpn = 0;
for (int i = 0; i < q; i++)
```

```

if (L <= ql[qpp[i]] && ql[qpp[i]] <= R)
    qp[qpn++] = qpp[i];

int ptr = R + 1;
for (int i = 0; i < qpn; i++) {
    int j = qp[i];
    while (ptr <= qr[j]) {
        int v = pp(a[ptr]);
        int u = pp(b[ptr]);
        int w = -1;
        if (v == n && u == n) {

        } else if (v == n) {
            ST.erase(pval[u]);
            p[u] = n;
            if (in_block[u])
                w = u;
        } else if (u == n) {
            ST.erase(pval[v]);
            p[v] = n;
            if (in_block[v])
                w = v;
        } else if (v != u) {
            if (in_block[u])
                swap(v, u);
            ST.erase(min(pval[v], pval[u]));
            if (in_block[v])
                ST.erase(max(pval[v], pval[u]));
            pval[v] = max(pval[v], pval[u]);
            p[u] = v;
            if (in_block[u])
                w = u;
        } else {
            ST.erase(pval[v]);
            p[v] = n;
            if (in_block[v])
                w = v;
        }
    }
    if (w != -1) {
        int p = cp[w];
    }
}

```

```

        cx[p] = cx[cx_n - 1];
        cp[cx[p]] = p;
        cx_n--;
    }
    ptr++;
}
ans[j] = get(ql[j], min(qr[j], R));
}
}

```

```

void solve() {
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> a[i];
    for (int i = 0; i < n; i++)
        cin >> b[i];
    cin >> q;
    for (int i = 0; i < q; i++) {
        cin >> ql[i] >> qr[i];
        ql[i]--, qr[i]--;
        qpp[i] = i;
    }
    sort(qpp, qpp + q, [](int i, int j) {
        return qr[i] < qr[j];
    });
    for (int i = 0; i < n; i += K)
        solve(i, min(n, i + K) - 1);
    for (int i = 0; i < q; i++)
        cout << ans[i] << "\n";
}

```

```

int main() {
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
#endif
    ios_base::sync_with_stdio(false);
    int t = 1;
    // cin >> t;
    for (int i = 1; i <= t; i++) {
        // cout << "Case #" << i << endl;
    }
}

```

```
    solve();  
  }  
}
```

Задача D:

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

using namespace std;

typedef double db;
typedef long long ll;
typedef long double ld;
typedef unsigned int ui;
typedef unsigned long long ull;

typedef pair < db, db > pdd;
typedef pair < db, ld > pdl;
typedef pair < ld, db > pld;
typedef pair < ld, ld > ldp;

typedef pair < ll, ll > pll;
typedef pair < int, ll > pil;
typedef pair < ll, int > pli;
typedef pair < int, int > pii;

#define F first
#define S second

#define en end()
#define bg begin()

#define rev reverse
#define mp make_pair
#define pb push_back

#define y1 y1234567890
#define um unordered_map

#define all(x) x.bg, x.en
#define sz(x) (int)x.size()
#define len(x) (int)strlen(x)
```



```

#define sqr(x) ((x + 0ll) * (x))
#define sqrd(x) ((x + 0.0) * (x))

#define forn(i, n) for (int i = 1; i <= n; i++)

const ll inf = (ll)1e18;
const ll mod = (ll)1e9 + 7;

const db eps = (db)1e-9;
const db pi = acos(-1.0);

const int dx[] = {0, 0, 1, 0, -1};
const int dy[] = {0, 1, 0, -1, 0};

const int N = 500001;
const int Log = 60;
const ll lim = (ll)1e18;

ll a[N], ar[N];
vector < pii > qr[N];
int n, q, pw[N], last1[Log], last2[Log], last[Log][N], mx[Log], cur_mx[Log], ans[N];

int cnt[Log];

inline void solve() {
    scanf("%d %d", &n, &q);
    for (int i = 1; i <= n; i++) {
        scanf("%lld", &a[i]);
        while (!(a[i] & 1)) {
            pw[i]++;
            a[i] >>= 1;
        }
        ar[i] = a[i];
    }

    for (int i = 1; i <= q; i++) {
        int l, r;
        scanf("%d %d", &l, &r);
        qr[r].pb({l, i});
    }
}

```

```

sort(ar + 1, ar + 1 + n);
int m = unique(ar + 1, ar + 1 + n) - ar - 1;

memset(last, -1, sizeof(last));

for (int i = 1; i <= n; i++) {
    a[i] = lower_bound(ar + 1, ar + 1 + m, a[i]) - ar;
    last[pw[i]][a[i]] = i;

    memset(last1, -1, sizeof(last1));
    memset(last2, -1, sizeof(last2));
    for (int j = 1; j + pw[i] < Log && last[pw[i] + j][a[i]] != -1; j++)
        last1[j] = last[pw[i] + j][a[i]];
    for (int j = 1; pw[i] - j >= 0 && last[pw[i] - j][a[i]] != -1; j++)
        last2[j] = last[pw[i] - j][a[i]];

    int p1 = 0, p2 = 0;
    last1[0] = last2[0] = i;
    cur_mx[0] = i;
    while ((p1 + 1 < Log && last1[p1 + 1] != -1) || (p2 + 1 < Log && last2[p2 + 1] !=
-1)) {
        if (last1[p1 + 1] != -1 && (last2[p2 + 1] == -1 || last1[p1 + 1] > last2[p2 +
1]))
            p1++;
        else
            p2++;
        assert(p1 + p2 < Log);
        cur_mx[p1 + p2] = min(cur_mx[p1 + p2 - 1], min(last1[p1], last2[p2]));
    }
    for (int j = 0; j < Log; j++)
        mx[j] = max(mx[j], cur_mx[j]);

    for (auto x : qr[i]) {
        int l = x.F, id = x.S;
        for (int j = 1; j < Log; j++) {
            if (mx[j] >= l)
                ans[id] = j;
        }
    }
}

```

```

    }

//   for (int i = 1; i <= q; i++)
//       cnt[ans[i]]++;
//   for (int i = 30; i < Log; i++)
//       printf("%d : %d\n", i, cnt[i]);

    for (int i = 1; i <= q; i++)
        printf("%d\n", ans[i]);
}

int main() {
    //freopen(".in", "r", stdin);
    //freopen(".out", "w", stdout);
    //freopen(".err", "w", stderr);

    //mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

    //cin.tie(NULL);
    //cout.tie(NULL);
    //ios_base::sync_with_stdio(false);

    //cout << setprecision(10) << fixed;

    int T = 1;
    //cin >> T;
    while (T--)
        solve();

    //cerr << (clock() + 0.0) / CLOCKS_PER_SEC;

    return 0;
}

```

Задача E:

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

using namespace std;
typedef long long ll;
const int maxn = 1e6 + 100;
const int mod = (int)1e9+7;
vector<int> g[maxn];
int q, t;
int nxt[maxn];
int p[maxn];
int n;
int a[maxn];
vector< pair<int, int> > Q[maxn];
ll ans[maxn][2];
struct node {
    ll sz;
    int lev;
    ll sum_lev;
    ll sum_lev_sqr;
    ll sum_up_lev[3];
    ll ans_up;
    ll ans_up_sqr;
} ver[maxn];
int get_parent(int v) {
    if(p[v] == v) return v;
    return p[v] = get_parent(p[v]);
}
}
long long mult(long long a, long long b) {
    a %= mod;
    b %= mod;
    return a * b % mod;
}
}
long long sum(long long a, long long b) {
    return ((a + b) % mod + mod) % mod;
}
}
void add(long long &a, long long x) {
    a += x;
}
```

```

    a %= mod;
    if(a < 0) a += mod;
}
void dfs(int v, int p) {
    ver[v].lev = ver[p].lev + 1;
    ver[v].sz = 1;
    ver[v].sum_lev = ver[v].lev;
    ver[v].sum_lev_sqr = mult(ver[v].lev, ver[v].lev);
    ver[v].ans_up_sqr = sum(sum(ver[v].sum_up_lev[2], -mult(2 * ver[v].sum_up_lev[1],
ver[v].lev)), mult(ver[v].sum_up_lev[0], ver[v].sum_lev_sqr));
    ver[v].ans_up = sum(mult(ver[v].lev, ver[v].sum_up_lev[0]), -ver[v].sum_up_lev[1]);
    for(int to: g[v]) {
        ll cur = (v-to);
        for(int i = 0; i < 3; i++) {
            ver[to].sum_up_lev[i] = sum(ver[v].sum_up_lev[i], cur);
            cur = mult(cur, ver[v].lev);
        }
        dfs(to, v);
        add(ver[v].sz, ver[to].sz);
        add(ver[v].sum_lev, ver[to].sum_lev);
        add(ver[v].sum_lev_sqr, ver[to].sum_lev_sqr);
    }
}
ll pref[maxn];
ll pref2[maxn];
ll pref_lev[maxn];
ll pref_lev_sqr[maxn];

struct PREF {
    ll pref_a;
    ll pref_b;
    ll pref_a_sqr;
    ll pref_b_sqr;
    PREF add_pref_sum(int v) {
        PREF x = {pref_a, pref_b, pref_a_sqr, pref_b_sqr};
        add(x.pref_a, ver[v].sum_lev - (ver[v].lev - 1) * ver[v].sz);
        add(x.pref_b, mult(v, sum(ver[v].sum_lev, -mult((ver[v].lev - 1), ver[v].sz))));
        add(x.pref_b, sum(mult(-ver[v].sum_up_lev[1], ver[v].sz), mult(ver[v].sum_up_lev[0],
ver[v].sum_lev)));
        ll cur = sum(sum(mult(mult((ver[v].lev-1), (ver[v].lev-1)), ver[v].sz),

```

```

        -mult(2 * ver[v].sum_lev, (ver[v].lev-1))), ver[v].sum_lev_sqr);
    add(x.prefa_sqr, cur);
    add(x.prefb_sqr, mult(cur, v));
    add(x.prefb_sqr, mult(ver[v].sum_up_lev[2], ver[v].sz)
        - mult(2 * ver[v].sum_lev, ver[v].sum_up_lev[1]) + mult(ver[v].sum_lev_sqr,
ver[v].sum_up_lev[0]));
    return x;
}
    PREF sub_pref_sum(PREF o) {
        PREF x = {sum(prefa, - o.prefa), sum(prefb, -o.prefb), sum(prefa_sqr, -o.prefa_sqr),
sum(prefb_sqr, -o.prefb_sqr)};
        return x;
    }
} b[maxn];
ll stu[maxn][2];
void solve() {
    cin >> n >> q >> t;
    t--;
    vector<int> st;
    for(int i = 1; i <= n; i++) {
        p[i] = i;
        cin >> a[i];
    }
    a[n+1] = 1<<<30;
    for(int i = 0; i < q; i++) {
        int l, r;
        cin >> l >> r;
        Q[r].push_back({l, i});
    }
    for(int i = 1; i <= n + 1; i++) {
        while(st.size() > 0 && a[i] > a[st.back()]) {
            g[i].push_back(st.back());
            st.pop_back();
        }
        st.push_back(i);
    }
    dfs(n+1, 0);

    for(int i = 1; i <= n; i++) {
        pref[i] = sum(pref[i-1], ver[i].ans_up);
    }
}

```

```

    pref2[i] = sum(pref2[i-1], ver[i].ans_up_sqr);
    pref_lev[i] = sum(pref_lev[i-1], ver[i].lev);
    pref_lev_sqr[i] = sum(pref_lev_sqr[i-1], mult(ver[i].lev, ver[i].lev));
}
st.clear();
for(int i = 1; i <= n; i++) {
    while(st.size() > 0 && a[i] > a[st.back()]) {
        p[st.back()] = i;
        st.pop_back();
    }
    st.push_back(i);

    b[st.size()] = b[st.size() - 1].add_pref_sum(i);
    for(auto [to, id]: Q[i]) {
        int v = get_parent(to);
        int pos = lower_bound(st.begin(), st.end(), v) - st.begin();
        PREF d = b[st.size()].sub_pref_sum(b[pos + 1]);
        ans[id][0] = sum(pref[i], -pref[to-1] + mult(d.prefa, (i+1)) - d.prefb);
        ans[id][1] = sum(pref2[i], -pref2[to-1] + mult(d.prefa_sqr, (i+1)) - d.prefb_sqr);
        {
            ll cur_lev = pref_lev[v] - pref_lev[to-1];
            ll sz = v - to + 1;
            ll cur = mult(cur_lev, ver[v].sum_up_lev[0]) - mult(ver[v].sum_up_lev[1], sz);
            add(ans[id][0], -cur);
            add(ans[id][0], mult((i+1 - v), (cur_lev - mult((ver[v].lev - 1), sz))));
        }
        {
            ll cur_lev = pref_lev[v] - pref_lev[to-1];
            ll sz = v - to + 1;
            ll cur_lev_sqr = pref_lev_sqr[v] - pref_lev_sqr[to - 1];
            ll cur = mult(ver[v].sum_up_lev[2], sz) - mult(2 * ver[v].sum_up_lev[1], cur_lev) +
mult(cur_lev_sqr, ver[v].sum_up_lev[0]);
            add(ans[id][1], -cur);
            ll m = mult(mult((ver[v].lev-1), (ver[v].lev-1)), sz) - mult(2 * (ver[v].lev - 1),
cur_lev) + cur_lev_sqr;
            add(ans[id][1], (i+1-v) * m);
        }
    }
}
}

```

```
    for(int i = 0; i < q; i++) {  
        cout << sum(ans[i][t], 0) << "\n";  
    }  
}  
int main() {  
    ios_base::sync_with_stdio(false);  
    cin.tie(0);  
    cout.tie(0);  
    int t = 1;  
    //cin >> t;  
    for(int i = 1; i <= t; i++) {  
        solve();  
    }  
}
```


Звдача F:

```
/**
 *   author:      fractal
 **/

#include <bits/stdc++.h>
using namespace std;

#define F first
#define S second
#define mp make_pair
#define pb push_back
#define pf push_front
#define ppb pop_back
#define ppf pop_front
#define speed ios_base::sync_with_stdio(0),cin.tie(0),cout.tie(0)
#define sz(x) (int)x.size()
#define len(x) (int)strlen(x)
#define all(x) x.begin(), x.end()
#define debug cerr << "OK\n";
#define ub upper_bound
#define lb lower_bound
#define make_unique(x) sort(all(x)), x.erase(unique(all(x)), x.end())

mt19937 bruh(chrono::steady_clock::now().time_since_epoch().count());
mt19937_64 rofl(chrono::steady_clock::now().time_since_epoch().count());

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef vector<pii> vpii;
typedef vector<pll> vpll;
typedef set<int> si;
typedef set<ll> sll;
typedef set<pii> spii;
```

```

typedef set<pll> spll;
typedef multiset <int> msi;
typedef multiset <ll> msl;
typedef map <int, int> mi;
typedef map <ll, ll> mll;

const int N = 1e6 + 2;
const int M = 4e5 + 500;
const int mod = 1e9 + 7;
const int inf = 2e9 + 3;
const ll INF = 1e18;
const ld pi2 = 2.0 * 3.141592653589793;
const ld pi = 3.141592653589793;
const ld eps = 1e-12;

const int dx[4] = {1, -1, 0, 0};
const int dy[4] = {0, 0, -1, 1};

#ifdef PC
    #define gcd __gcd
#endif

void files(string s = "main") {
    #ifdef PC
        freopen((s + ".in").c_str(), "r", stdin);
        freopen((s + ".out").c_str(), "w", stdout);
    #endif
}

int add(int a, int b) {
    if (a + b < 0) return a + b + mod;
    if (a + b >= mod) return a + b - mod;
    return a + b;
}

int mul(int a, int b) {
    return a * 1LL * b % mod;
}

int binpow(int a, int n) {

```

```

int ret = 1;
while (n) {
    if (n & 1) ret = mul(ret, a);
    a = mul(a, a);
    n >>= 1;
}
return ret;
}

int fact[N], invf[N];

int C(int n, int k) {
    if (n < 0 || k > n) return 0;
    return mul(fact[n], mul(invf[k], invf[n - k]));
}

int all_ways(int a, int b, int c, int d) {
    return C(d - b + c - a, c - a);
}

int good_ways(int a, int b, int c, int d) {
    b--, d--;
    return add(all_ways(a, b, c, d), -all_ways(b, a, c, d));
}

int n, p[3][N];
int ans = 1;
vpri v;

int main() {
    speed;
    fact[0] = 1;
    for (int i = 1; i <= M; ++i) {
        fact[i] = mul(fact[i - 1], i);
    }
    invf[M] = binpow(fact[M], mod - 2);
    for (int i = M; i >= 1; --i) {
        invf[i - 1] = mul(invf[i], i);
    }
    cin >> n;
}

```

```

for (int i = 1; i <= n; ++i) {
    cin >> p[1][i];
    if (p[1][i] != 0) {
        v.pb({i, p[1][i] - i});
        v.pb({i - 1, p[1][i] - i});
    }
}
for (int i = 1; i <= n; ++i) {
    cin >> p[2][i];
    if (p[2][i] != 0) {
        v.pb({p[2][i] - i, i});
        v.pb({p[2][i] - i, i - 1});
    }
}
sort(all(v));
// for (auto [x, y] : v)
//     cerr << x << " " << y << '\n';
pii last = {0, 0};
for (auto it : v) {
    if (it.F < it.S) {
        cout << 0 << '\n';
        return 0;
    }
    if (last.F <= it.F && last.S <= it.S) {
        ans = mul(ans, good_ways(last.F, last.S, it.F, it.S));
        last = it;
        continue;
    }
    cout << 0 << '\n';
    return 0;
}
ans = mul(ans, good_ways(last.F, last.S, n, n));
cout << ans << '\n';
}

```