DS

莫队

```
int n, m, len;
const int N = 200010;

bool cmp(array<int, 3> a, array<int, 3> b) {
    int x = a[0]/len;
    int y = b[0]/len;
    if (x != y) return x < y;
    else return a[1] < b[1];
}

void solve() {
    cin >> n >> m;
    vec<int> a(n+1,0);
    vec<int> cnt(N, 0);
    vec<array<int, 3>> evt;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    for (int i = 1; i <= m; i++) {
        int l, r;
        cin >> l >> r;
        evt.pb({l, r, i});
    }
    len = max(1ll, (ll)sqrtl(1.0 * n * n / m));
    sort(all(evt), cmp);

    ll ans = 0;
    int l = 0, r= 0;
    ll res[N];

    auto add = [&](int p) {
        cnt[a[p]]++;
        if (cnt[a[p]] == 2) ans++;
    };

    auto sub = [&](int p) {
```

```
        cnt[a[p]]--;
        if (cnt[a[p]] == 1) ans--;
    };


    for (auto x : evt) {
        while (r < x[1]) r++, add(r);
        while (r > x[1]) sub(r), r--;
        while (l < x[0]) sub(l), l++;
        while (l > x[0]) l--, add(l);
        res[x[2]] = ans;
    }
    for (int i = 1; i <= m; i++) {
        cout << (res[i]==0?"Yes":"No") << '\n';
    }
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    int _;
    _ = 1;
    for (int i = 1; i <= _; i++) {
        solve();
    }
    return 0;
}
```

树状数组(求逆序对)

```
// https://atcoder.jp/contests/abc296/submissions/40293683
const int N = 200010;
int c[N];
int n;
void modify (int x, int y) {
    for (x; x <= n; x += x & -x) c[x] += y;
}
int query(int x) {
    int s = 0;
    for (x; x; x -= x & -x) s += c[x];
    return s;
}

void solve(int Tcase = 1) {
    cin >> n;
    vec<int> a(n), b(n);

    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }

    for (int i = 0; i < n; i++) {
        cin >> b[i];
```

```
    }

    ll res1 = 0, res2 = 0;
    for (int i = n - 1; i >= 0; i--) {
        modify(a[i], 1);
        res1 += query(a[i] - 1);
    }

    for (int i = 1; i <= n; i++) c[i] = 0;

    for (int i = n - 1; i >= 0; i--) {
        modify(b[i], 1);
        res2 += query(b[i] - 1);
    }

    sort(a.begin(), a.end());
    sort(b.begin(), b.end());

    bool f = 0;
    for (int i = 0; i < n; i++) {
        if (a[i] != b[i]) {
            cout << "No\n";
            return ;
        }
        if (i && a[i] == a[i - 1]) {
            f = 1;
        }
    }

    if (f) {
        cout << "Yes\n";
        return ;
    }

    cout << ((res1+res2)%2==0? "Yes":"No");
}
```

拓扑排序判环

```
// https://atcoder.jp/contests/abc296/submissions/40269554
const int N = 200010;
int n;
vec<int> e[N];
void solve(int Tcase = 1) {
    cin >> n;
    vec<int> a(n + 1, 0);
    vec<int> idg(n + 1, 0);
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        idg[a[i]]++;
        e[i].pb(a[i]);
    }

    queue<int> que;
    for (int i = 1; i <= n; i++) {
```

```
        if (!idg[i]) que.push(i);
    }
    int ans = n;

    while (!que.empty()) {
        int t = que.front();
        que.pop();
        ans--;
        for (auto v : e[t]) {
            if (--idg[v] == 0) {
                que.push(v);
            }
        }
    }
    cout << ans << '\n';
}
```

树链剖分

```
// 树链剖分
const int N = 200010;
int n, m, a[N];
int dep[N], l[N], r[N], top[N], id[N], e_to_p[N];
int son[N], sz[N], fa[N], tot, p_to_e[N];
// l[x], r[x]求出的是第x个点在dfs序上的区间范围
// e_to_p[x] 求的是第x条边依附于哪个点
// p_to_e[x] 求的是第x个点的边是哪个
vec<P(int)> e[N];
void dfs1(int u, int f) {
    fa[u] = f;
    dep[u] = dep[f] + 1;
    son[u] = -1;
    sz[u] = 1;
    for (auto [v, p]: e[u]) {
        if (v == f) continue;
        dfs1(v, u);
        e_to_p[p] = v;
        p_to_e[v] = p;
        sz[u] += sz[v];
        if (son[u] == -1 || sz[son[u]] < sz[v]) {
            son[u] = v;
        }
    }
}

void dfs2(int u, int t) {
    top[u] = t;
    l[u] = ++tot;
    id[tot] = u;
    if (son[u] != -1) {
        dfs2(son[u], t);
    }
    for (auto [v, p]: e[u]) {
        if (v == fa[u] || v == son[u]) continue;
        dfs2(v, v);
```

```
    }
    r[u] = tot;
}

struct seg{
    ll sum;
    seg operator + (const seg& R) const {
        return (seg){sum + R.sum};
    }
    seg operator - (const seg& R) const {
        return (seg){sum - R.sum};
    }
} tr[N << 4];

void pushup(int p) {
    tr[p] = tr[lp] + tr[rp];
}

void build(int p, int l, int r) {
    if (l == r) {
        tr[p] = {1ll * a[p_to_e[id[l]]]};
    } else {
        int mid = (l + r) / 2;
        build(lp, l, mid);
        build(rp, mid + 1, r);
        pushup(p);
    }
}

void modify(int p, int l, int r, int pos, ll d) {
    if (l == r) {
        tr[p].sum = d;
    } else {
        int mid = (l + r) / 2;
        if (pos <= mid) modify(lp, l, mid, pos, d);
        else modify(rp, mid + 1, r, pos, d);
        pushup(p);
    }
}

seg query(int p, int l, int r, int ql, int qr) {
    if (ql <= l && r <= qr) return tr[p];
    int mid = (l + r)/2;
    if (ql > mid)
        return query(rp, mid + 1, r, ql, qr);
    else if (qr <= mid)
        return query(lp, l, mid, ql, qr);
    else
        return query(lp, l, mid, ql, qr) +
        query(rp, mid + 1, r, ql, qr);
}

seg query(int u, int v) {
    seg ans = {0};
    int p = u, q = v;
    while (top[u] != top[v]) {
```

```
        if (dep[top[u]] < dep[top[v]]) {
            if (u != v)
                ans = ans + query(1, 1, n, l[top[v]], l[v]);
            v = fa[top[v]];
        } else {
            if (u != v)
                ans = ans + query(1, 1, n, l[top[u]], l[u]);
            u = fa[top[u]];
        }
    }
    if (v == u) return ans;
    if (dep[v] < dep[u]) {
        ans = ans + query(1, 1, n, l[v]+1, l[u]);
    } else {
        ans = ans + query(1, 1, n, l[u]+1, l[v]);
    }
    return ans;
}

void solve(int caseT = 1) {
    cin >> n;
    for (int i = 1; i < n; i++) {
        int u, v, w;
        cin >> u >> v >> w;
        e[u].pb({v, i});
        e[v].pb({u, i});
        a[i] = w;
    }
    dfs1(1, 0);
    dfs2(1, 1);
    build(1, 1, n);
    cin >> m;
    for (int i = 1; i <= m; i++) {
        int op, x, y;
        cin >> op >> x >> y;
        if (op == 1) {
            modify(1, 1, n, l[e_to_p[x]], y);
        } else {
            seg ans = query(x, y);
            cout << ans.sum << '\n';
        }
    }
}
```

DSU on Tree

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
const int N = 100010;
int hs[N], sz[N], l[N], r[N], tot, id[N];
vector<int> e[N];
int cnt[N];
```

```
int a[N], n;
ll sumcnt, maxcnt, ans[N];
void dfs(int u, int fa) {
    sz[u] = 1;
    hs[u] = -1;
    l[u] = ++tot;
    id[tot] = u;
    for (auto v : e[u]) {
        if (v == fa) continue;
        dfs(v, u);
        sz[u] += sz[v];
        if (hs[u] == -1 || sz[v] > sz[hs[u]])
            hs[u] = v;
    }
    r[u] = tot;
}

void DSU(int u, int fa, bool keep) {
    for (auto v : e[u]) {
        if (v != fa && v != hs[u])
            DSU(v, u, false);
    }
    if (hs[u] != -1) {
        DSU(hs[u], u, true);
    }

    auto add = [&](int x) {
        int col = a[x];
        cnt[col]++;
        if (cnt[col] > maxcnt)
            maxcnt = cnt[col], sumcnt = col;
        else if (cnt[col] == maxcnt)
            sumcnt += col;
    };

    auto del = [&](int x) {
        int col = a[x];
        cnt[col]--;
    };

    for (auto v : e[u]) {
        if (v != fa && v != hs[u]) {
            for (int x = l[v]; x <= r[v]; x++)
                add(id[x]);
        }
    }
    add(u);
    ans[u] = sumcnt;
    if (!keep) {
        maxcnt = 0;
        sumcnt = 0;
        for (int x = l[u]; x <= r[u]; x++) {
            del(id[x]);
        }
    }
}
```

```
int main () {
    scanf("%d", &n);
    for (int i = 1; i <= n; i++) scanf("%d", a + i);
    for (int i = 1; i < n; i++) {
        int u, v;
        scanf("%d%d", &u, &v);
        e[u].push_back(v);
        e[v].push_back(u);
    }
    dfs(1, 0);
    DSU(1, 0, false);
    for (int i = 1; i <= n; i++)
        printf("%lld%c", ans[i], " \n"[i == n]);
    return 0;
}
```

根号分治

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
const int N = 200010, M = 500;
int n, m, q;
ll tag[M + 10][M + 10], val[N];
void solve() {
    cin >> n >> q;
    for (int i = 0;i < q; i++) {
        int op, x;
        cin >> op >> x;
        if (op == 1) {
            int y, d;
            cin >> y >> d;
            if (x <= M) {
                tag[x][y] += d;
            } else {
                for (int j = y; j <= n; j += x) {
                    val[j] += d;
                }
            }
        } else {
            ll ans = val[x];
            for (int j = 1; j <= M; j++) {
                ans += tag[j][x%j];
            }
            cout << ans << '\n';
        }
    }
}
```

对拍

随机数生成

```
mt19937_64 rnd(time(0));
ll a = rnd();
```

对拍代码

```
#include<bits/stdc++.h>
using namespace std;
int main() {
    for (int i = 1; i <= 1000; i++) {
        system("rand > a.in"); // linux下 ./rand
        system("ac < a.in > ac.out"); // linux下 ./ac
        system("wa < a.in > wa.out"); // linux下 ./wa
        if (system("fc ac.out wa.out")) { // // linux下 diff
            printf("Case %i is Wa!!!\n", i);
            exit(0);
        } else {
            printf("Case %i is Ac!!!\n", i);
        }
    }
}
```

INT128

```
#include <bits/stdc++.h>
#define P __int128
using namespace std;

P read() {
    P s = 0, f = 1;
    char c = getchar();
    while (c < '0' || c > '9') {
        if (c == '-') f = -1;
        c = getchar();
    }
    while ('0' <= c && c <= '9') {
        s = s * 10 + c - '0';
        c = getchar();
    }
    return s * f;
}
void write(P x) {
    if (x >= 10) write(x/10);
    putchar(x%10+'0');
}
int main() {
    P a = read();
    write(a);}
```