主席树_昆明M

主席树_区间第k小

主席树_区间前K大的和

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

#define fi first
#define se second
#define mp make_pair
#define lowbit(x) ((x) & -(x))
#define ls (p << 1)
#define rs (ls | 1)
#define tm ((tl + tr) >> 1)

using namespace std;
using ll = long long;
using ull = unsigned long long;
using pii = pair<int, int>;

constexpr double eps = 1e-8;
constexpr int NINF = 0xc0c0c0c0;
constexpr int INF = 0x3f3f3f3f;
constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
constexpr ll mod = 1e9 + 7;
constexpr ll N = 2e6 + 5;

struct line {
    int l, r, h, op;

    inline bool operator<(const line &T) const {
        return h == T.h ? op > T.op : h < T.h;
    }
} a[N];

int n, m, x[N], cnt[N << 2], len[N << 2];

void push_up(int p, int tl, int tr) {
    if (cnt[p]) {
        len[p] = x[tr] - x[tl];
    } else {
        len[p] = len[ls] + len[rs];
    }
}

```

```cpp
void modify(int p, int tl, int tr, int L, int R, int v) {
    if (x[tr] <= L || R <= x[tl]) return;
    if (L <= x[tl] && x[tr] <= R) {
        cnt[p] += v;
        push_up(p, tl, tr);
        return;
    }
    if (L <= x[tm]) modify(ls, tl, tm, L, R, v);
    if (R > x[tm]) modify(rs, tm, tr, L, R, v);
    push_up(p, tl, tr);
}

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

    cin >> n;
    for (int i = 0; i < n; i++) {
        int x1, y1, x2, y2;
        cin >> x1 >> y1 >> x2 >> y2;
        a[++m] = {x1, x2, y1, 1};
        x[m] = x1;
        a[++m] = {x1, x2, y2, -1};
        x[m] = x2;
    }
    n = m;
    sort(a + 1, a + 1 + n);
    sort(x + 1, x + 1 + n);
    m = unique(x + 1, x + 1 + n) - (x + 1);
    ll ans = 0;
    for (int i = 1; i < n; i++) {
        modify(1, 1, m, a[i].l, a[i].r, a[i].op);
        ans += 1ll * len[1] * (a[i + 1].h - a[i].h);
    }
    cout << ans << '\n';

    return 0;
}
```

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

#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define ls (p << 1)
```

```cpp
#define rs (ls | 1)
#define tm ((tl + tr) >> 1)
#define lowbit(x) ((x) & -(x))

using namespace std;
using db = double;
using ll = long long;
using ull = unsigned long long;
using pii = pair<int, int>;

const db PI = acos(-1.0);
constexpr db EPS = 1e-8;
constexpr int NINF = 0xc0c0c0c0;
constexpr int INF = 0x3f3f3f3f;
constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
constexpr ll MOD = 1e9 + 7;
constexpr ll N = 1e6 + 5;

int tot, rt[N];

struct node {
    int l, r;
    ll sum;
} t[N << 5];

void modify(int &x, int l, int r, int v) {
    t[++tot] = t[x];
    x = tot;
    t[x].sum += v;
    if (l == r) return;
    int mid = (l + r) >> 1;
    if (v <= mid) modify(t[x].l, l, mid, v);
    else modify(t[x].r, mid + 1, r, v);
}

ll query(int x, int y, int l, int r, ll v) {
    if (l == r) return t[y].sum - t[x].sum;
    int mid = (l + r) >> 1;
    if (v <= mid) return query(t[x].l, t[y].l, l, mid, v);
    else return query(t[x].r, t[y].r, mid + 1, r, v) + t[t[y].l].sum -
t[t[x].l].sum;
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout << fixed << setprecision(20);

```

```
    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++) {
        int x;
        cin >> x;
        modify(rt[i] = rt[i - 1], 1, 1000000000, x);
    }
    int l, r;
    ll ans = 0;
    while (m--) {
        cin >> l >> r;
        l = (l + ans) % n + 1;
        r = (r + ans) % n + 1;
        if (l > r) swap(l, r);
        ans = 0;
        while (true) {
            ll res = query(rt[l - 1], rt[r], 1, 1000000000, ans + 1);
            if (res == ans) break;
            ans = res;
        }
        cout << ++ans << '\n';
    }

    return 0;
}
```

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

#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define ls (p << 1)
#define rs (ls | 1)
#define tm ((tl + tr) >> 1)
#define lowbit(x) ((x) & -(x))

using namespace std;
using db = double;
using ll = long long;
using ull = unsigned long long;
using pii = pair<int, int>;

const db PI = acos(-1.0);
constexpr db EPS = 1e-8;
constexpr int NINF = 0xc0c0c0c0;
constexpr int INF = 1e9;
constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
```

```cpp
constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
constexpr ll MOD = 1e9 + 7;
constexpr ll N = 1e6 + 5;

int rt[N], a[N], tot;

struct node {
    int l, r, cnt;
} t[N << 5];

void modify(int &x, int l, int r, int v) {
    t[++tot] = t[x];
    x = tot;
    t[x].cnt++;
    if (l == r) return;
    int mid = (l + r) >> 1;
    if (v <= mid) modify(t[x].l, l, mid, v);
    else modify(t[x].r, mid + 1, r, v);
}

int query(int x, int y, int l, int r, int v) {
    if (l == r) return l;
    int mid = (l + r) >> 1, cnt = t[t[y].l].cnt - t[t[x].l].cnt;
    if (v <= cnt) return query(t[x].l, t[y].l, l, mid, v);
    else return query(t[x].r, t[y].r, mid + 1, r, v - cnt);
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout << fixed << setprecision(20);

    int n, m;
    cin >> n >> m;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        modify(rt[i] = rt[i - 1], -INF, INF, a[i]);
    }
    while (m--) {
        int l, r, k;
        cin >> l >> r >> k;
        cout << query(rt[l - 1], rt[r], -INF, INF, k) << '\n';
    }

    return 0;
}
```

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

```cpp

#define fi first
#define se second
#define mp make_pair
#define pb push_back
#define ls (p << 1)
#define rs (ls | 1)
#define tm ((tl + tr) >> 1)
#define lowbit(x) ((x) & -(x))

using namespace std;
using db = double;
using ll = long long;
using ull = unsigned long long;
using pii = pair<int, int>;

const db PI = acos(-1.0);
constexpr db EPS = 1e-8;
constexpr int NINF = 0xc0c0c0c0;
constexpr int INF = 1000000;
constexpr ll LNINF = 0xc0c0c0c0c0c0c0c0;
constexpr ll LINF = 0x3f3f3f3f3f3f3f3f;
constexpr ll MOD = 1e9 + 7;
constexpr ll N = 1e5 + 5;

struct node {
    int l, r, cnt;
    ll sum;
} t[N << 5];

int tot, rt[N];

void modify(int &x, int l, int r, int v) {
    t[++tot] = t[x];
    x = tot;
    t[x].cnt++;
    t[x].sum += v;
    if (l == r) return;
    int mid = (l + r) >> 1;
    if (v <= mid) modify(t[x].l, l, mid, v);
    else modify(t[x].r, mid + 1, r, v);
}

ll query(int x, int y, int l, int r, int v) {
    if (l == r) return 1ll * v * l;
    int mid = (l + r) >> 1, cnt = t[t[y].r].cnt - t[t[x].r].cnt;
    if (v <= cnt) return query(t[x].r, t[y].r, mid + 1, r, v);
    else return query(t[x].l, t[y].l, l, mid, v - cnt) + t[t[y].r].sum -
t[t[x].r].sum;
```

```cpp
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    cout << fixed << setprecision(20);

    int T;
    cin >> T;
    while (T--) {
        tot = 0;
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++) {
            int x;
            cin >> x;
            modify(rt[i] = rt[i - 1], 1, 1000000, x);
        }
        int m;
        cin >> m;
        while (m--) {
            int l, r, k;
            cin >> l >> r >> k;
            int x = r - l + 1;
            cout << 1ll * x * (x + 1) * (2 * x + 1) / 6 + query(rt[l - 1], rt[r], 1, 1000000, k) << '\n';
        }
    }

    return 0;
}
```