



树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

树状数组套权值线段树

河南省实验中学信息技术组

2026年04月17日



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

给定一个含有 n 个数的序列 $a_1, a_2 \dots a_n$ ，需要支持两种操作：

- Q l r k ，表示查询下标在区间 $[l, r]$ 中的第 k 小的数；
- C x y ，表示将 a_x 改为 y 。

数据范围： $1 \leq n, m \leq 10^5, 0 \leq y \leq 10^9$ 。



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

【输入格式】

第一行两个正整数 n, m ，表示序列长度与操作个数。

第二行 n 个整数，表示 $a_1, a_2 \dots a_n$ 。

接下来 m 行，每行表示一个操作，都为上述两种中的一个。

【输出格式】

对于每个查询操作，输出一行一个整数表示答案。

【样例输入】

```
5 3
3 2 1 4 7
Q 1 4 3
C 2 6
Q 2 5 3
```

【样例输出】

```
3
6
```



【例】动态第 k 小的数

- 对于查询，直接对区间 $a[l, r]$ 内的数排序，然后输出第 k 小的数即可。
- 对于修改操作，直接修该即可。
- 时间复杂度： $O(MN \log N)$ 。

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

- 如果不考虑修改，可以用可持久化权值线段树来解决。
- 可持久化权值线段树中的第 i 棵线段树保存的是 $a[1 : i]$ 中每个数出现的次数，每次查询 $a[l : r]$ 中的数出现的次数，需要用第 r 棵线段树减去第 $l - 1$ 棵线段树，类似于前缀和。
- 考虑修改，可以用维护动态前缀和的方法来维护这些线段树，即将线段树看成一个整体放在树状数组中。
- 对于将 a_x 修改为 y ，会影响第 x 棵及以后的所有线段树，用树状数组维护时只需要影响第 $x, x + \text{lowbit}(x) \dots$ 棵线段树。
- 因为 a_i, y 的数据范围较大，需要先将这些数离散化。



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

```
1 // 此处 p 为引用 主要是为了方便新建线段树
2 void update(int &p, int L, int R, int x, int y)
3 {
4     if(!p) p = ++tot;
5     if(L == R) { cnt[p] += y; return ; }
6     if(x <= M) update(lc[p], L, M, x, y);
7     else update(rc[p], M + 1, R, x, y);
8     cnt[p] = cnt[lc[p]] + cnt[rc[p]];
9 }
10
11 // 主函数中调用 add(i, a[i], 1)
12 void add(int x, int y, int z)
13 {
14     for(int i = x; i <= n; i += i & -i)
15         update(rt[i], 1, s, y, z); // s 是离散化后的数据范围
16 }
```



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

```
1 // 建立初始可持久化线段树
2 for(int i = 1; i <= n; ++i)
3 {
4     a[i] = lower_bound(b + 1, b + s + 1, a[i]) - b; // 离散化
5     add(i, a[i], 1); // 在第 i 棵线段树上进行单点修改 a[i] 出现的次数加 1
6 }
7 // 单点修改 a[x] 改为 y
8 y[i] = lower_bound(b + 1, b + s + 1, y[i]) - b; // 离散化
9 add(x[i], a[x[i]], -1); // 原来 a[x[i]] 出现次数减 1
10 a[x[i]] = y[i];
11 add(x[i], a[x[i]], 1); // 新数 a[x[i]] 出现次数加 1
```



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

- 对于查询区间 $[l, r]$ 的第 k 小的数，显然需要处理前 r 棵权值线段树和前 $l-1$ 棵权值线段树，同样用树状数组维护时只需要统计第 $r, r - \text{lowbit}(r), \dots$ 棵线段树和第 $l-1, l-1 - \text{lowbit}(l-1), \dots$ 棵线段树。
- 那么可以将 r 查询到的所有线段树合并起来，将 $l-1$ 查询到的线段树合并起来，然后相减就是目前 $a[l, r]$ 中所有数的权值线段树。
- 但是上述操作不方便，可以将 r 和 $l-1$ 需要统计的线段树都记录下来，例如 $l-1$ 的线段树都记为 $p[]$ ， r 的线段树都记为 $q[]$ ，那么每次只计算时，首先统计所有 $p[]$ 的左孩子元素个数之和 A 和所有 $q[]$ 的左孩子元素个数之和 B ：
 - 如果 $k \leq B - A$ ，那么说明第 k 小的数在左子树中，所有的 $p[]$ 和 $q[]$ 都移动到左孩子；
 - 否则，说明第 k 小的数在右子树中，且为 $k - (B - A)$ 小的数，所有的 $p[]$ 和 $q[]$ 都移动到右孩子。
- 时间复杂度： $O(N \log^2 N)$ ，空间复杂度： $O(N \log N \log 2N)$ 。



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

```
1 // 此查询的树的结点分别存储在  $p[]$  和  $q[]$  中
2 int query(int p[], int q[], int L, int R, int k)
3 {
4     if(L == R) return L;
5     int lcnt = 0; // 所有左子树中出现的数的差 即出现在  $[L[i], r[i]]$  中的数的个数
6     for(int i = 1; i <= tp; ++i) lcnt -= cnt[lc[p[i]]];
7     for(int i = 1; i <= tq; ++i) lcnt += cnt[lc[q[i]]];
8     if(k <= lcnt) // 第  $k$  小的数在左孩子中
9     {
10        for(int i = 1; i <= tp; ++i) p[i] = lc[p[i]];
11        for(int i = 1; i <= tq; ++i) q[i] = lc[q[i]];
12        return query(p, q, L, M, k);
13    }
14    else // 第  $k$  小的数在右孩子中
15    {
16        for(int i = 1; i <= tp; ++i) p[i] = rc[p[i]];
17        for(int i = 1; i <= tq; ++i) q[i] = rc[q[i]];
18        return query(p, q, M + 1, R, k - lcnt); // 查询第  $k - lcnt$  小的数
19    }
20 }
```



【例】动态第 k 小的数

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

```
1 int ask(int l, int r, int k)
2 {
3     tp = tq = 0;
4     for(int i = l - 1; i; i -= i & - i) p[+tp] = rt[i]; // l-1 对应的线段树
5     for(int i = r; i; i -= i & - i) q[+tq] = rt[i]; // r 对应的线段树
6     return query(p, q, 1, s, k);
7 }
8 // 主函数
9 int x = ask(l[i], r[i], k[i]);
10 printf("%d\n", b[x]); // 离散化前的数
```



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

【题目描述】

有 n 个元素，第 i 个元素有 a_i, b_i, c_i 三个属性，设 $f(i)$ 表示满足 $a_j \leq a_i$ 且 $b_j \leq b_i$ 且 $c_j \leq c_i$ 且 $j \neq i$ 的 j 的数量。

对于所有 $d \in [0, n)$ ，求 $f(i) = d$ 的数量。

【输入格式】

第一行两个整数 n, k ，表示元素数量和最大属性值。

接下来 n 行，每行三个整数 a_i, b_i, c_i ，分别表示三个属性值。

【输出格式】

共 n 行，第 $d + 1$ 行表示 $f(i) = d$ 的 i 的数量。

【数据范围】

$1 \leq n \leq 10^5$ ， $1 \leq a_i, b_i, c_i \leq k \leq 2 \times 10^5$



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

【样例输入】

```
10 3
3 3 3
2 3 3
2 3 1
3 1 1
3 1 2
1 3 1
1 1 2
1 2 2
1 3 2
1 2 1
```

【样例输出】

```
3
1
3
0
1
0
1
0
0
1
```



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

- 算法 1: 模拟
- 时间复杂度: $O(N^2)$ 。



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

- 考虑二维偏序 (a_i, b_i) ，一般按照 b_i 从小到大排序，用权值树状数组或权值线段树对前 i 个元素的 a_i 值的情况。
- 当枚举元素 (a_i, b_i) 时，统计权值线段树中 $[1, a_i]$ 的元素个数，然后将 a_i 插入权值线段树即可。
- 对于三维偏序 (a_i, b_i, c_i) ，先按照 c_i, b_i 和 a_i 从小到大排序，用树状数组维护 a_i ，用线段树维护 b_i 。
- 当枚举元素 (a_i, b_i, c_i) 时，利用树状数组统计第 $\dots, a_i - \text{lowbit}(a_i), a_i$ 线段树中值域在 $[1, b_i]$ 的元素个数；然后在树状数组的 $a_i, a_i + \text{lowbit}(a_i), \dots$ 线段树中的 b_i 处加 1。
- 注意：对于 c_i 相等的元素要先插入到树状数组中，然后再查询答案，避免漏算。
- 时间复杂度： $O(N \log^2 N)$ ，空间复杂度： $O(2N \log 2N \log 2N)$ 。



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

```
1 void update(int &p, int L, int R, int x, int y)
2 {
3     if(!p) p = ++tot;
4     if(L == R) { cnt[p] += y; return; }
5     if(x <= M) update(lc[p], L, M, x, y);
6     else update(rc[p], M + 1, R, x, y);
7     cnt[p] = cnt[lc[p]] + cnt[rc[p]];
8 }
9
10 void add(int x, int y, int z)
11 {
12     for(int i = x; i <= k; i += i & -i)
13         update(rt[i], 1, k, y, z);
14 }
```



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

```
1 int query(int p[], int L, int R, int x) // p[] 为多个线段树
2 {
3     if(L == R) {
4         int res = 0;
5         for(int i = 1; i <= t; ++i) res += cnt[p[i]];
6         return res;
7     }
8     if(x <= M) {
9         for(int i = 1; i <= t; ++i) p[i] = lc[p[i]];
10        return query(p, L, M, x);
11    } else{
12        int sum = 0;
13        for(int i = 1; i <= t; ++i) sum += cnt[lc[p[i]]], p[i] = rc[p[i]];
14        return sum += query(p, M + 1, R, x);
15    }
16 }
17
18 int ask(int x, int y)
19 {
20     t = 0;
21     for(int i = x; i; i -= i & -i) p[++t] = rt[i];
22     return query(p, 1, k, y);
23 }
```



【例】三维偏序

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第k小的数
三维偏序

练习

```
1 sort(e + 1, e + n + 1);
2 for(int i = 1; i <= n;)
3 {
4     int j = i; // 先将所有 c[i] 相等的元素插入
5     while(e[j].c == e[j + 1].c) add(e[j].a, e[j].b, 1), ++j;
6     add(e[j].a, e[j].b, 1);
7     j = i; // 统计对应的结果
8     while(e[j].c == e[j + 1].c) ans[e[j].id] = ask(e[j].a, e[j].b), ++j;
9     ans[e[j].id] = ask(e[j].a, e[j].b);
10    i = j + 1;
11 }
12 for(int i = 1; i <= n; ++i) ++d[ans[i] - 1]; // 注意删除自己本身
13 for(int i = 0; i <= n - 1; ++i) printf("%d\n", d[i]);
```



练习

树状数组套权
值线段树

河南省实验中学
信息技术组

例题

动态第 k 小的数
三维偏序

练习

- 动态第 k 小的数 (COGS 4332)
- 平衡树 (树套树)(COGS 1594)
- 三维偏序 (COGS 3336)
- 序列 [HEOI/TJOI 2016](COGS 3326)
- 动态逆序对 [CQOI2011](COGS 1715)
- 任务查询系统 [CQOI 2015](COGS 1936)
- 路灯 [APIO 2019](COGS 3384)