

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme Ringo by memset0
Proudly powered by Typecho

洛谷4647 [IOI2007] sails 船帆

2019-03-04 | 题解

容易发现答案与旗杆的顺序无关，我们可以把旗杆按照从矮到高的顺序排序，这样的话我们只需要维护两个操作：

1. 插入一些 0
2. 给一个前缀整体 +1

考虑每次 +1 操作，除了这个前缀中的最大值，每个数的相对位置不会改变。考虑放到线段树上，对于最大值二分出其区间，对其特殊处理后移即可。

代码：

```
// =====  
//   author: memset0  
//   date: 2019.03.03 19:23:30  
//   website: https://memset0.cn/  
// =====  
#include <bits/stdc++.h>  
#define ll long long  
namespace ringo {  
template <class T> inline void read(T &x) {  
    x = 0; register char c = getchar(); register bool f = 0;  
    while (!isdigit(c)) f ^= c == '-', c = getchar();  
    while (isdigit(c)) x = x * 10 + c - '0', c = getchar();  
}
```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memseto](#)
Proudly powered by [Typecho](#)

```

    if (f) x = -x;
}
template <class T> inline void print(T x) {
    if (x < 0) putchar('-'), x = -x;
    if (x > 9) print(x / 10);
    putchar('0' + x % 10);
}
template <class T> inline void print(T x, char c) { print(x), putchar(c); }

const int N = 1e5 + 10;
int n, m; ll ans;

struct node {
    int h, k;
    inline bool operator < (const node &other) const {
        return h < other.h;
    }
} a[N];

struct seg_node {
    int l, r, mid, sum;
} p[N << 2];

void build(int u, int l, int r) {
    p[u].l = l, p[u].r = r, p[u].mid = (l + r) >> 1;
    if (l == r) { return; }
    build(u << 1, l, p[u].mid), build(u << 1 | 1, p[u].mid + 1, r);
}

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

int query(int u, int k) {

```



memseto's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

关于我
友情链接
文章聚合

Theme [Ringo](#) by [memseto](#)
Proudly powered by [Typecho](#)

```

    if (p[u].l == p[u].r) return p[u].sum;
    return query(k <= p[u].mid ? u << 1 : u << 1 | 1, k) + p[u].sum;
}

int query_bound(int _l, int _r, int k) {
    int ans = _l - 1;
    for (int l = _l, r = _r, mid; mid = (l + r) >> 1, l <= r; )
        if (query(1, mid) == k) ans = mid, r = mid - 1;
        else l = mid + 1;
    return ans;
}

int query_bound2(int _l, int _r, int k) {
    int ans = _l;
    for (int l = _l + 1, r = _r, mid; mid = (l + r) >> 1, l <= r; )
        if (query(1, mid) == k) ans = mid, l = mid + 1;
        else r = mid - 1;
    return ans;
}

void main() {
    read(n);
    for (int i = 1; i <= n; i++) read(a[i].h), read(a[i].k);
    std::sort(a + 1, a + n + 1), build(1, 1, m = a[n].h);
    for (int i = 1, pos, end, limit = m + 1, length; i <= n; i++) {
        limit -= a[i].h - a[i - 1].h;
        modify(1, limit, limit + a[i].k - 1, 1);
        if (limit + a[i].k - 1 < m && query(1, limit + a[i].k - 1) > query(1,
            pos = query_bound(limit, limit + a[i].k - 1, query(1, limit + a[i].k));
            end = query_bound2(limit + a[i].k, m, query(1, limit + a[i].k));
            length = limit + a[i].k - pos;
            modify(1, pos, pos + length - 1, -1);
            modify(1, end - length + 1, end, 1);
        }
    }
    for (int i = 1, x; i <= m; i++)
        x = query(1, i), ans += (1ll)x * (x - 1) / 2;
}

```



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

- 关于我
- 友情链接
- 文章聚合

Theme **Ringo** by **memset0**
Proudly powered by **Typecho**

```
print(ans, '\n');  
}  
  
} signed main() { return ringo::main(), 0; }
```

线段树

用户名
邮箱
网址（选填）
可以在这里写评论哦 ~
提交评论

洛谷4208 [JSOI2008]最小生成树计数
上一篇 «

在 lxd 中安装 SYZOJ
» 下一篇



memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

© 2017 - 2019 [memset0](#) 的博客.

[浙ICP备19006255号-1](#)

97711 visits · 24757 visitors · 74.48 W words

在这里输入关键字哦 ~ (回车搜索)

[关于我](#)

[友情链接](#)

[文章聚合](#)

Theme [Ringo](#) by [memset0](#)

Proudly powered by [Typecho](#)

