

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**

LJOJ5249 老夫

2019-03-01 | 题解

非常好的分块 + 凸包题。

讲每个人按照 b 排序并依次插入，题目可以转换为维护

- 插入一个数；
- 查询从大到小排序后每个数的排名乘以自身的最大值。

考虑分块维护插入操作

- 暴力重构当前块
- 单调栈维护下凸壳维护整体排名 + 1 操作，对于当前块之后的块，执行这个操作。

查询时直接查询即可。

代码：

```
// =====  
//   author: memset0  
//   date: 2019.02.28 15:26:37  
//   website: https://memset0.cn/  
// =====
```

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

```
#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();
    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, c, w, sqn, now, c_blk, max_a, max_b;

struct node { int a, b, bln, pos; } a[N];
struct line { int k, b; inline ll calc(int x) const { return (ll)x * k + b; }
inline bool cmp_a(const node &a, const node &b) { return a.a > b.a; }
inline bool cmp_b(const node &a, const node &b) { return a.b < b.b; }
inline double cross(const line &a, const line &b) { return (a.b - b.b) / (dou
inline double cross(double ak, double ab, double bk, double bb) { return (ab

struct block {
    int pos; ll ans;
    std::vector <node> vet, act;
    struct stack {
        std::vector <line> stk;
        inline int size() { return stk.size(); }
        inline line top() { return *--stk.end(); }
        inline line next_top() { return *----stk.end(); }
        inline void pop() { stk.pop_back(); }
        inline void clear() { stk.clear(); }
        inline void push(int k, int b) {
```



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**

```

        while (size() > 1)
            if (cross(stk[size() - 1], stk[size() - 2]) <= cross(k, b, to)
                else break;
        stk.push_back((line){k, b});
    }
} stk;
void move() {
    ++pos;
    while (stk.size() > 1) if (stk.top().calc(pos) < stk.next_top().calc(
        ans = stk.size() ? stk.top().calc(pos) : 0;
    }
void rebuild(const node &it) {
    act.push_back(it), vet = act, stk.clear();
    std::sort(vet.begin(), vet.end(), cmp_a);
    for (int i = 0; i < vet.size(); i++)
        stk.push(vet[i].a, vet[i].a * (i + 1));
    --pos, move();
}
void out() {
    putchar('{');
    for (auto it = stk.stk.begin(); it != stk.stk.end(); it++)
        printf("[%d %d]", it->k, it->b);
    putchar('}');
}
} blk[N];

void extend(const node &it) {
    for (int i = it.blk + 1; i <= c_blk; i++) blk[i].move();
    blk[it.blk].rebuild(it);
}

ll solve() {
    ll ans = 0;
    for (int i = 1; i <= c_blk; i++) ans = std::max(ans, blk[i].ans);
    return ans + (ll)c * w * (n - now + 1);
}

```



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**

```
void main() {
    read(n), read(w), sqn = sqrt(n);
    for (int i = 1; i <= n; i++) read(a[i].a), read(a[i].b);
    std::sort(a + 1, a + n + 1, cmp_a), max_a = a[1].a;
    for (int i = 1; i <= n; i++) a[i].pos = i, a[i].bln = (i - 1) / sqn + 1;
    c_blk = (n - 1) / sqn + 1;
    std::sort(a + 1, a + n + 1, cmp_b), max_b = a[n].b;
    for (c = 1, now = 1; c <= max_b + 1; c++) {
        while (now <= n && a[now].b < c) extend(a[now]), ++now;
        print(solve(), ' ');
    }
}

} signed main() { return ringo::main(), 0; }
```

分块

单调栈

已有 2 条评论



daniel14311531

March 1st, 2019 at 02:02 pm

然而我被卡常了QAQ

回复



Isonan

March 1st, 2019 at 01:57 pm

%%%



[回复](#)

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](#)

用户名
邮箱
网址（选填）
可以在这里写评论哦 ~
提交评论

[洛谷2496 \[SDOI2012\]体育课](#)
上一篇 «

[LOJ6287 诗歌](#)
» 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

97716 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**

