

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2496 [SDOI2012]体育课

2019-03-01 | 题解

与上一题相同的思路，考虑分块 + 凸包来维护。

- 对于 1 操作，两端的块暴力查询，中间的取整块计算的结果
- 对于 2 操作，暴力重构两个块
- 对于 3 操作，两端的暴力重构，中间的同样用单调栈来维护。

代码：

```
// =====
//  author: memset0
//  date: 2019.03.01 10:05:27
//  website: https://memset0.cn/
// =====
#include <bits/stdc++.h>
#define ll long long
#define int 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, M = 500;
int n, m, sqn;
int a[N], bln[N];

struct line {
    int k, b;
    inline int calc(int x) {
        return k * x + b;
    }
};

inline double cross(const line &a, const line &b) {
    return (a.b - b.b) / (double)(b.k - a.k);
}

struct block {
    int id, tim, base, pos, ans, fst, end;
    std::vector<line> stk;
    void push(const line &u) {
        while (stk.size() > 1)
            if (cross(*--stk.end(), u) > cross(*--stk.end(), *----stk.end()))
                stk.pop_back();
            else break;
        stk.push_back(u);
    }
    void play() {
        for (int i = 1; i <= end - fst + 1; i++)
            a[fst + i - 1] += i * tim + base;
        tim = base = 0;
    }
};

```



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**

```

}
void move(int dis) {
    tim += dis, pos += dis;
    while (stk.size() > 1)
        if ((--stk.end())->calc(pos) < (----stk.end())->calc(pos))
            stk.pop_back();
        else break;
    ans = (--stk.end())->calc(pos) + base;
}
void rebuild() {
    ans = pos = 0, stk.clear();
    for (int i = fst; i <= end; i++)
        ans = std::max(ans, a[i]);
    for (int i = end; i >= fst; i--)
        push((line){i - fst + 1, a[i]});
}
void modify(int l, int r, int x, int base = 0) {
    play();
    for (int i = l; i <= r; i++) a[i] += x * (i - l + 1) + base;
    rebuild();
}
int query(int l, int r) {
    play(); int ans = 0;
    for (int i = l; i <= r; i++) ans = std::max(ans, a[i]);
    rebuild(); return ans;
}
} b[M], *blk[N];

void modify(int l, int r, int x) {
    if (bln[l] == bln[r]) blk[l]->modify(l, r, x);
    else {
        for (int i = bln[l] + 1; i < bln[r]; i++)
            b[i].base += (b[i].fst - l) * x, b[i].move(x);
        blk[l]->modify(l, blk[l]->end, x);
        blk[r]->modify(blk[r]->fst, r, x, (blk[r]->fst - l) * x);
    }
}
}

```



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

```
int query(int l, int r) {
    int ans = 0;
    if (bln[l] == bln[r]) ans = blk[r]->query(l, r);
    else {
        for (int i = bln[l] + 1; i < bln[r]; i++)
            ans = std::max(ans, b[i].ans);
        ans = std::max(ans, blk[l]->query(l, blk[l]->end));
        ans = std::max(ans, blk[r]->query(blk[r]->fst, r));
    } return ans;
}

void swap(int l, int r) {
    blk[l]->play(), blk[r]->play();
    std::swap(a[l], a[r]);
    blk[l]->rebuild(), blk[r]->rebuild();
}

void main() {
    read(n), read(m), sqn = sqrt(n);
    for (int i = 1; i <= n; i++) {
        read(a[i]);
        bln[i] = (i - 1) / sqn + 1;
        blk[i] = &b[bln[i]];
    }
    for (int i = n; i >= 1; i--) blk[i]->fst = i;
    for (int i = 1; i <= n; i++) blk[i]->end = i;
    for (int i = 1; i <= bln[n]; i++) b[i].id = i, b[i].rebuild();
    for (int i = 1, l, r, x, op; i <= m; i++) {
        read(op), read(l), read(r);
        if (op == 1) print(std::max(query(l, r) - a[l] - b[l].tim - b[l].base);
        if (op == 2) swap(l, r);
        if (op == 3) read(x), modify(l, r, x);
    }
}

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



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**

[巧妙的思路](#)[分块](#)[单调栈](#)

可以在这里写评论哦 ~

CF538G Berserk Robot
上一篇 «

LJOJ5249 老夫
» 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](#)

97715 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](#)

