memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

Theme Ringo by memseto

Proudly powered by Typecho

BZOJ5016 [SNOI2017]一个简单的询问

2019-02-24 │ 题解

把询问拆分后直接莫队。

考虑

$$
\begin{aligned}
\sum_{i=0}^{\infty} get(l_1, r_1, x) \times get(l_2, r_2, x) = &\sum_{i=0}^{\infty} get(0, r_1, x) \times get(0, r_2, x) \\
&- \sum_{i=0}^{\infty} get(0, l_1 - 1, x) \times get(0, r_2, x) \\
&- \sum_{i=0}^{\infty} get(0, r_1, x) \times get(0, l_2 - 1, x) \\
&+ \sum_{i=0}^{\infty} get(0, l_1 - 1, x) \times get(0, l_2 - 1, x)
\end{aligned}
$$

一个比较通俗的理解是 $(A - C)(B - D) = AB - BC - AD + CD$ 。

代码：

```cpp
// ================================
//    author: memset0
//    date: 2019.02.24 22:34:44
//    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();
    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 = 5e4 + 10;
int n, m, tn, ul, ur, ql, qr, qc, sqn;
int a[N], b[N], L[N], R[N], bln[N];
ll now, ans[N];

struct query {
    int l, r, mul, id;
    inline bool operator < (const query &other) const {
        return bln[l] == bln[other.l] ? r < other.r : l < other.l;
    }
} q[N << 2];

inline void update(int k, int w, int *A) {
    now -= (ll)L[k] * R[k], A[k] += w, now += (ll)L[k] * R[k];
}
```

memset0's
Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

Theme Ringo by memseto

Proudly powered by Typecho

```cpp
inline void modify(int l, int r, int w, int *A) {
    for (int i = l; i <= r; i++) update(a[i], w, A);
    // printf("modify %d %d %d => %lld\n", l, r, w, now);
}


void main() {
    read(n), sqn = sqrt(n);
    for (int i = 1; i <= n; i++) read(a[i]), b[++tn] = a[i];;
    std::sort(b + 1, b + tn + 1), tn = std::unique(b + 1, b + tn + 1) - b - 1
    for (int i = 1; i <= n; i++) b[i] = std::lower_bound(b + 1, b + tn + 1, a
    for (int i = 1; i <= n; i++) bln[i] = (i - 1) / sqn + 1;
    read(m);
    for (int i = 1, l1, l2, r1, r2; i <= m; i++) {
        read(l1), read(r1), read(l2), read(r2);
        q[++qc] = (query){r1, r2,  1, i}, q[++qc] = (query){l1 - 1, l2 - 1,
        q[++qc] = (query){l1 - 1, r2, -1, i}, q[++qc] = (query){r1, l2 - 1, -
        // modify(1, r1,  1, L), modify(1, r2,  1, R);
        // ans[i] += now;
        // printf(">>> %d\n", now);
        // modify(1, r1, -1, L), modify(1, r2, -1, R);

        // modify(1, l1 - 1,  1, L), modify(1, r2,  1, R);
        // ans[i] -= now;
        // printf(">>> %d\n", now);
        // modify(1, l1 - 1, -1, L), modify(1, r2, -1, R);

        // modify(1, r1,  1, L), modify(1, l2 - 1,  1, R);
        // ans[i] -= now;
        // printf(">>> %d\n", now);
        // modify(1, r1, -1, L), modify(1, l2 - 1, -1, R);

        // modify(1, l1 - 1,  1, L), modify(1, l2 - 1,  1, R);
        // ans[i] += now;
        // printf(">>> %d\n", now);
        // modify(1, l1 - 1, -1, L), modify(1, l2 - 1, -1, R);
    }
```

↑

```cpp
    std::sort(q + 1, q + qc + 1), ul = 0, ur = 0;
    for (int i = 1; i <= qc; i++) {
        ql = q[i].l, qr = q[i].r;
        while (ul < ql) update(a[++ul], 1, L); while (ul > ql) update(a[ul--]
        while (ur < qr) update(a[++ur], 1, R); while (ur > qr) update(a[ur--]
        ans[q[i].id] += q[i].mul * now;
    }
    for (int i = 1; i <= m; i++) print(ans[i], '\n');
}

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

巧妙的思路   莫队

用户名

邮箱

网址（选填）

可以在这里写评论哦 ~

提交评论

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

memset0's Notebook

方知蓦然回首之时
那人却已不在灯火阑珊处

多项式复合逆学习笔记
上一篇 «

MySQL 数据库基本操作学习笔记
» 下一篇

© 2017 - 2019 memset0 的博客.
浙ICP备19006255号-1
97719 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 memseto

Proudly powered by Typecho