莫队配合bitset

bitset 常用于常规数据结构难以维护的的判定、统计问题，而莫队可以维护常规数据结构难以维护的区间信息。把两者结合起来使用可以同时利用两者的优势。

例题 「Ynoi2016」掉进兔子洞 [https://www.luogu.com.cn/problem/P4688]

本题刚好符合上面提到的莫队配合 bitset 的特征。不难想到我们可以分别用 bitset 存储每一个区间内的出现过的所有权值，一组询问的答案即所有区间的长度和减去三者的并集元素个数 $\times 3$ 。

但是在莫队中使用 bitset 也需要针对 bitset 的特性调整算法：

1. bitset 不能很好地处理同时出现多个权值的情况。我们可以把当前元素离散化后的权值与当前区间的的出现次数之和作为往 bitset 中插入的对象。
2. 我们平常使用莫队时，可能会不注意 4 种移动指针的方法顺序，所以指针移动的过程中可能会出现区间的左端点在右端点右边，区间长度为负值的情况，导致元素的个数为负数。这在其他情况下并没有什么影响，但是本题中在 bitset 中插入的元素与元素个数有关，所以我们需要注意 4 种移动指针的方法顺序，将左右指针分别往左边和右边移动的语句写在前面，避免往 bitset 中插入负数。
3. 虽然 bitset 用空间小，但是仍然难以承受 $10^5 \times 10^5$ 的数据规模。所以我们需要将询问划分成常数块分别处理，保证空间刚好足够的情况下时间复杂度不变。

参考代码

```
#include <algorithm>
#include <bitset>
#include <cmath>
```

```cpp
#include <cstdio>
#include <cstring>
using namespace std;
const int N = 100005, M = N / 3 + 10;
int n, m, maxn;
int a[N], ans[M], cnt[N];
bitset<N> sum[M], now;
struct query {
  int l, r, id;
  bool operator<(const query& x) const {
    if (l / maxn != x.l / maxn) return l < x.l;
    return (l / maxn) & 1 ? r < x.r : r > x.r;
  }
} q[M * 3];
void static_set() {
  static int tmp[N];
  memcpy(tmp, a, sizeof(a));
  sort(tmp + 1, tmp + n + 1);
  for (int i = 1; i <= n; i++)
    a[i] = lower_bound(tmp + 1, tmp + n + 1, a[i]) - tmp;
}
void add(int x) {
  now.set(x + cnt[x]);
  cnt[x]++;
}
void del(int x) {
  cnt[x]--;
  now.reset(x + cnt[x]);
}
void solve() {
  int cnt = 0, tot = 0;
  now.reset();
  for (tot = 0; tot < M - 5 && m; tot++) {
    m--;
    ans[tot] = 0;
    sum[tot].set();
    for (int j = 0; j < 3; j++) {
      scanf("%d%d", &q[cnt].l, &q[cnt].r);
      q[cnt].id = tot;
      ans[tot] += q[cnt].r - q[cnt].l + 1;
      cnt++;
    }
  }
  sort(q, q + cnt);
  for (int i = 0, l = 1, r = 0; i < cnt; i++) {
    while (l > q[i].l) add(a[--l]);
    while (r < q[i].r) add(a[++r]);
    while (l < q[i].l) del(a[l++]);
    while (r > q[i].r) del(a[r--]);
```

```
53      sum[q[i].id] &= now;
54    }
55    for (int i = 0; i < tot; i++)
56      printf("%d\n", ans[i] - (int)sum[i].count() * 3);
57  }
58  int main() {
59    scanf("%d%d", &n, &m);
60    for (int i = 1; i <= n; i++) scanf("%d", &a[i]);
61    static_set();
62    maxn = sqrt(n);
63    solve();
64    memset(cnt, 0, sizeof(cnt));
65    solve();
66    memset(cnt, 0, sizeof(cnt));
67    solve();
68    return 0;
69  }
```

习题

- 小清新人渣的本愿 [https://www.luogu.com.cn/problem/P3674]
- 「Ynoi2017」由乃的玉米田 [https://www.luogu.com.cn/problem/P5355]
- 「Ynoi2011」WBLT [https://www.luogu.com.cn/problem/P5313]
