

1 setmod

1.1 30%

$O(nm)$ 暴力。

1.2 100%

要维护几个东西：

- `sum` 表示区间的和
- `type` 表示现在的标记类型（可以是没有标记，可以是增量标记，可以是赋值标记）
- `delta` 如果是增量标记，那么这个里面存的增量
- `value` 如果是赋值标记，那么这里面就存的是那个值

然后就秩序要讨论一下发现：

- 空标记 + 赋值操作 = 赋值标记
- 空标记 + 增量操作 = 增量标记
- 增量标记 + 赋值操作 = 赋值标记
- 增量标记 + 增量操作 = 增量标记
- 赋值标记 + 增量操作 = 赋值标记
- 赋值标记 + 赋值操作 = 赋值标记

因为它们之间能和谐共处（如果 A 标记遇见 B 操作无法转化为一种现有的标记，那么他们就不能同时用线段树实现），所以就能完成两种操作同时进行。具体看代码是怎样合并的。

2 area

2.1 30%

枚举点，判断它是否在某个矩形中。 $O(n^3)$

2.2 100%

扫描线，对于一个举行 $(x1,y1,x2,y2)$ ，将它看成两个事件：在 $x1$ 这个时间将 $(y1,y2)$ 这个区间加一，在 $x2+1$ 这个时间将 $(y1,y2)$ 这个区间减一。

这样，我们遍历整个时间，并在执行完这个时间的操作后看看有多少位置非 0，将其数量加到答案中，就完了，当然时间不能傻傻地一个一个枚，因为关键的时间点最多 $2n$ 个，其它时候面积是没有变的，所以要一段一段地算。

至于怎么用线段树实现那么查看有多少个非零的位置，需要注意对于任何一个减一操作，前面一定有一个和它一样的加一操作，就只需要维护一下每个节点被完全覆盖的次数。再用另一个来统计子树中的那些修改导致这个节点还有多少个非零。有点像标记永久化（我们讲的第二种区间修改的写法）。

3 intkth

3.1 30%

暴力, sort, $O(mn \log n)$

3.2 100%

树状数组套可持久化值域线段树, 每次询问就是对 $O(\log n)$ 棵树加加减减后求第 k 大。修改也会同时修改 $O(\log n)$ 棵树, 所以总的复杂度是: $O((m+n) \log^2 n)$ 。这道题我记得课上讲过的哟。