

题目描述

你有一个 n 行 m 列的网格，每个格子可以被染黑或染白，初始每个格子均为白色。定义第 i 行的美观程度为最大的 x ，满足对于所有 $1 \leq j \leq x$ ，第 i 行的第 j 个格子的颜色是黑色。

你需要对这个网格执行 m 次操作。操作有以下四种：

1. 给定三个整数 l, r, x ，表示把第 x 列的第 l 到第 r 个格子染黑。
2. 给定三个整数 l, r, x ，表示把第 x 列的第 l 到第 r 个格子染白。
3. 给定三个整数 l, r, v ，你需要把第 l 到第 r 行中美观程度最小的行的权值加上 v ，如有多行均为最小则对每行都加 v 。
4. 给定两个整数 l, r ，你需要输出第 l 到第 r 行的权值之和，对 2^{64} 取模。

每行初始权值为 0。

输入格式

第一行包含两个整数 n, m 。

接下来 m 行，每行第一个正整数 op_i 表示操作编号。

对于 $op_i = 1$ 或 $op_i = 2$ ，后面紧跟三个正整数 l_i, r_i, x_i ，表示执行上述对应操作。

对于 $op_i = 3$ ，后面紧跟三个整数 l_i, r_i, v_i ，表示执行上述对应操作。

对于 $op_i = 4$ ，后面紧跟两个整数 l_i, r_i ，表示询问这个区间的权值之和对 2^{64} 取模的值。

输出格式

若干行，对于每个 $op_i = 4$ 输出其答案。

样例一

input

```
3 5
1 2 2 1
3 1 3 1
4 1 1
4 1 2
4 1 3
```

output

```
1
1
2
```

数据规模与约定

对于 100% 的数据，

$1 \leq n, m \leq 3 \times 10^5, 1 \leq l_i \leq r_i \leq n, 1 \leq x_i \leq \min(m, 1.5 \times 10^5), 0 \leq v_i < 2^{64}$ 。

本题采用子任务捆绑测试。

subtask1(10pts): 保证 $1 \leq n, m \leq 1000$ 。

subtask2(15pts): 保证 $x_i = 1$ 。

subtask3(15pts): 保证 $1 \leq x_i \leq 10$ 。

subtask4(20pts): 保证 $op_i = 3$ 和 $op_i = 4$ 的操作一定出现在所有 $op_i = 1$ 和 $op_i = 2$ 的操作之后。

subtask5(10pts): 保证 $1 \leq n, m \leq 40000$ 。

subtask6(30pts): 无特殊性质。

时间限制: 4s

空间限制: 1024MB

解题过程

算法一

对每个点开一个线段树, 可以在 $\log(m)$ 时间内修改和查询一行的美观程度, 复杂度 $O(nm \log m)$ 。

期望得分 10 分。

算法二

对整个序列开一个线段树, 维护每个格子的颜色, 支持区间染黑/染白, 区间白格子权值加 v , 区间权值加 v , 查询区间和。复杂度 $O(m \log n)$ 。

期望得分 15 分, 结合算法一能获得 25 分。

算法三

开 10 颗算法二中的线段树维护即可, 复杂度 $O(10 \times m \log n)$ 。

期望得分 30 分, 结合算法一能获得 40 分。

算法四

我们可以对每一列开一个 set 维护这一列上黑色格子的连续段, 在所有 $op_i = 1$ 和 $op_i = 2$ 的操作结束后通过扫描线线段树得到每一行的美观值。我们建一颗线段树, 每个节点上维护区间最小美观度, 区间最小美观度出现次数, 区间权值和, 区间最小美观度的权值被加了多少。注意到这个 tag 是支持下放的, 只需要在下放时判断左右孩子是不是区间最小值, 这样就能高效进行操作了, 复杂度 $O(m \log \max(n, m))$ 。

期望得分 20 分, 结合算法一和算法三能获得 60 分。

算法五

注意到算法四中的线段树是可以拓展到动态情况的, 我们使用线段树套 set 来维护每个点的美观值, 一行的美观值可以看作这行编号最小的白色格子。把白色格子的连续段插入线段树中, 每行的美观值就是从叶子节点到根路径上 set 中的最小值, 用和算法四类似的标记即可, 下放时需要注意一些细节, 复杂度 $O(m \log n \log m)$ 。

这里的 set 其实只需要支持插入/删除/查询最小值, 可以用可删堆维护, 常数会小很多。

期望得分 100 分, 如果常数较大只能获得 70 分。

参考资料

1. 2020-2021 Winter Petrozavodsk Camp, Day 5, [Rectangle Painting](#).