
混凝土粉末 (concrete)

【题目描述】

namespace_std 是一名 Minecraft 玩家。

这天，他开 op 拿到了无数的混凝土粉末，准备扔进一个世界里。

这个世界可以视为一个长度为 n ，宽度为 1，高度无限的平面，其中 $(x, 0)$ 的位置为基岩，其上方均为空气。

namespace_std 会进行两种操作共计 q 次：

1 $l\ r\ h$: 在区间 $[l, r]$ 内每个横坐标上方无限高处扔下 h 块第 x 种混凝土粉末，其中 x 是这次操作的序号。混凝土粉末被扔下后会一直下落直到落到另一个方块上方。

2 $x\ y$: 查询 (x, y) 位置是哪种混凝土。如果 (x, y) 上此时是空气输出 0。

【输入格式】

从文件 *concrete.in* 读入数据。

输入的第一行是两个整数 n, q ，分别表示这个世界的长度和操作的次数。

接下来 q 行，每行三至四个数，表示一次操作，意义见【题目描述】。

【输出格式】

输出到 *concrete.out* 中。

对于每次 2 操作，输出一行表示该组询问的答案。

【输入样例 1】

见选手目录下的 *concrete/concrete1.in*。

```
5 8
1 1 4 2
2 3 1
2 3 3
1 2 5 1
2 3 3
2 5 2
2 1 2
2 1 3
```

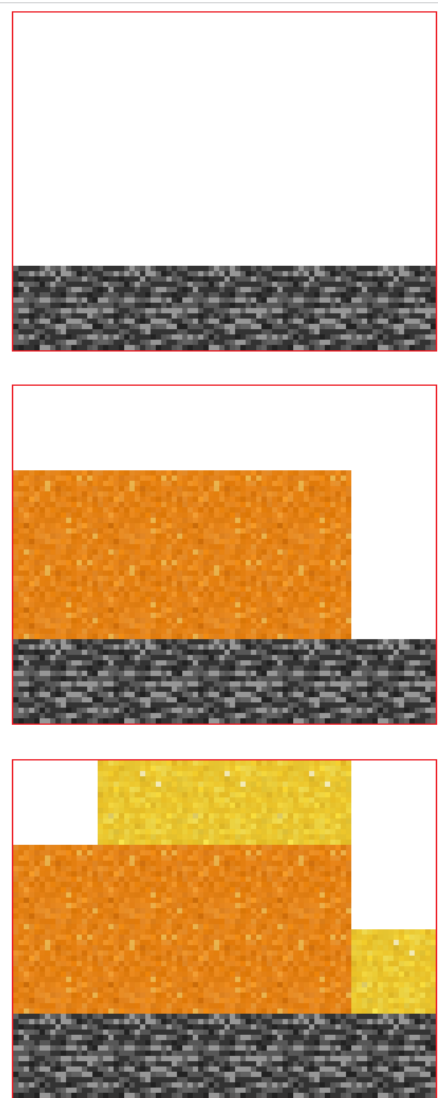
【输出样例 1】

见选手目录下的 *concrete/concrete1.ans*。

1
0
4
0
1
0

【样例解释 1】

两次操作后的平面如下图所示。



【输入样例 2】

见选手目录下的 *concrete/concrete2.in*。

```
10 15
1 5 7 2
1 2 9 3
1 1 10 1
1 1 2 2
1 4 8 1
1 1 6 1
2 3 5
2 7 2
2 8 6
2 5 7
2 2 6
2 10 1
2 9 3
2 1 3
2 3 1
```

【输出样例 2】

见选手目录下的 *concrete/concrete2.ans*。

```
6
1
0
5
4
3
2
4
2
```

【样例 3】

见选手目录下的 *concrete/concrete3.in* 和 *concrete/concrete3.ans*。

【数据规模与约定】

对于 10% 的数据， $1 \leq n, q \leq 100$ 。

对于 30% 的数据, $1 \leq n, q \leq 2000$ 。

对于另外 15% 的数据, 保证所有 2 操作在 1 操作之后。

对于另外 20% 的数据, $h = 1$ 。

对于 80% 的数据 (包含上述特殊性质), $1 \leq n, q \leq 10^5$ 。

对于 100% 的数据, $1 \leq n, q \leq 10^6, 1 \leq l \leq r \leq n, 1 \leq h \leq 10^9, 1 \leq x \leq n, 1 \leq y \leq 10^{18}$ 。