

卡片整理

时间限制：1.0 秒

空间限制：16 MiB

相关文件：题目目录



题目描述

小 M 有 n 摞卡片，最初每一摞中都只有一张卡片。现在小 M 想对这些卡片进行 m 步整理，每步整理为：取出某一摞卡片最上面的一张，输出这张卡片的编号，然后放到另外一摞的最上面。

整理的过程中，肯定会出现某些摞一张卡片都没有的情况。小 M 不会试图从没有卡片的卡片摞中取出卡片。

经过努力，小 M 终于完成了卡片摞的整理。但正在学习数据结构的小 M 还想再挑战一下自己：他把一部分整理步骤改成了询问某一摞卡片第 k 张的编号。——我们认为这道题满分就是 80 分，挑战部分可以不完成。

你需要帮助小 M 完成卡片摞的整理和新的挑战。

输入格式

第一行包含 2 个正整数 n 和 m ，表示卡片摞的数目、整理和询问的总数。其中卡片摞的编号为 1 到 n 的整数，每摞卡片中最初的卡片编号与卡片摞的编号相同。

接下来 m 行，每行描述一个整理步骤或一个询问，其中每行首先会包含 1 个整数 1 或 2。

若为 1，表示一个整理步骤，接下来会有两个整数 x 和 y ，表示把编号为 x 的卡片摞最上面的一张卡片移动到编号为 y 的卡片摞最上面，同时输出这张卡片的编号。（保证卡片摞编号合法，保证 $x \neq y$ ，且不会试图移动空的卡片摞）

若为 2，表示一个询问，接下来会有两个整数 x 和 k ，表示需要输出编号为 x 的卡片摞从下往上数第 k 张（从 1 开始数）卡片的编号。（保证 k 不大于这摞卡片的张数）

输出格式

输出 m 行，每行一个整数，表示每个整理步骤或询问对应的卡片编号。

样例输入

5 20
1 1 5
2 5 1
1 5 2
2 2 2
1 3 5
2 5 2
1 4 5
2 5 1
1 2 5
2 5 2
1 5 4
2 4 1
1 5 3
2 3 1
1 5 3
2 3 1
1 5 4
2 4 2
1 2 3
2 3 3

样例输出

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

子任务

- 1. 有 40% 的数据, $n \leq 1000$, $m \leq 10000$, 且只有整理卡片, 没有询问。

- 2. 有 40% 的数据, $n \leq 100000$, $m \leq 1000000$, 且只有整理卡片, 没有询问。
- 3. 有 10% 的数据, $n \leq 1000$, $m \leq 10000$ 。
- 4. 有 10% 的数据, $n \leq 100000$, $m \leq 1000000$ 。

在最终计时时, 80 分将视为满分。

提示

对于第 2 个子任务, 我们无法开 $n \times m$ 大小的二维数组, 但是 n 和 m 的若干倍空间是开得下的。我们学了哪些可以用来存储“总元素个数一定”的“二维数组”的方法呢?

对于第 4 个子任务, 链表是难以胜任的, 因为数据可能反复查找同一擦, 而这一擦卡片数又可能非常多。那么还有哪些存储“总元素个数一定”的“二维数组”的方法呢?

语言及编译选项信息

| # | 名称 | 编译器 | 额外参数 | 代码长度限制 (B) |
|---|----------------|---------|-------------------------------|------------|
| 0 | g++ with std11 | g++ | -O2 -std=c++11 -DONLINE_JUDGE | 65536 |
| 1 | g++ | g++ | -O2 -DONLINE_JUDGE | 65536 |
| 2 | gcc with std11 | gcc | -O2 -std=c11 -DONLINE_JUDGE | 65536 |
| 3 | gcc | gcc | -O2 -DONLINE_JUDGE | 65536 |
| 4 | java | javac | | 65536 |
| 5 | python | python | | 65536 |
| 6 | python3 | python3 | | 65536 |

递交历史

| # | 状态 | 时间 |
|----------------------------|----|----|
| No data available in table | | |

递交答案

语言和编译选项

g++ with std11

▼

1

递交评测

文件请拖入编辑器中，或

上传文件