

CSP2023模拟赛

题目名称	载客电梯	智慧之神	矩阵入门	MEX 求和
题目类型	传统型	传统型	传统型	传统型
可执行文件名	elevator	wisdom	matrix	mex
输入文件名	elevator.in	wisdom.in	matrix.in	mex.in
输出文件名	elevator.out	wisdom.out	matrix.out	mex.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限	512 MiB	512 MiB	512 MiB	512 MiB
子任务/测试点数目	6	20	10	6
是否等分	否	是	是	否

提交源文件程序名

对于C++语言	elevator.cpp	wisdom.cpp	matrix.cpp	mex.cpp
---------	--------------	------------	------------	---------

编译选项

对于C++语言	-lm -O2 -std=c++17
---------	--------------------

注意事项(请仔细阅读)

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++中函数main()的返回类型必须是int，程序正常结束时返回值必须是0。
- 选手提交的程序代码文件请在**个人目录下以及子文件夹内各放一份**。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于100KB。
- 程序可使用的栈空间内存限制于题目的内存限制一直。
- 使用std::deque等STL容器时，请注意其内存空间消耗。
- 评测时采用的机器配置为 AMD Ryzen 7 5800H with Radeon Graphics，内存16GiB。上述时限以此配置为准。
- 评测在Windows 10下进行，使用LemonLine进行评测。

载客电梯 (elevator)

【题目描述】

小 Z 搬进了新机房，新机房在书鹤琴院，是幢很高的楼，所以小 Z 上下楼都用电梯。这幢楼只有一个电梯，核载人数为 k 。

某时刻小 Z 准备坐电梯下楼，此时电梯正在一楼，而他发现有 $n - 1$ 个人也要下楼（加上他一共 n 个，都要下到一楼）。

小 Z 通过黑进大楼的监控，得知了 n 个人分别在哪一层。

为了简单化这个问题，我们假定这个电梯开关门/上下人不计时间，上一层楼的时间为 1。

然而电梯的系统十分死板，只会一路向上到位置最高的人那里，再一路向下，然后再一次往返去接这一次没坐进电梯的人。

于是善良的小 Z 又黑进了电梯的系统，希望能通过调配电梯的运作使得所有人的总等待时间最少。

一个人的等待时间定义为从开始到他坐电梯到一楼经过的时间。

【输入格式】

第一行两个整数 n, k ，表示要坐电梯的人数和电梯荷载人数。

接下来一行 n 个整数，第 i 个整数 a_i 表示第 i 个人所在楼层。

【输出格式】

一行一个整数，表示最小的总等待时间。

【输入输出样例1】

elevator.in	elevator.out
6 6 2 2 5 6 2 5	42

【数据规模与约定】

本题采用捆绑测试。对于所有的测试数据，保证 $1 \leq k \leq n \leq 5 \times 10^5, 2 \leq a_i \leq 2 \times 10^6$ 。

子任务编号	特殊性质	分值
1	$n \leq 10$	15
2	$k \leq 20$	12
3	$a_i \leq 10$	16
4	$n \leq 1000$	28
5	$k = n$	19
6	无	10

智慧之神 (wisdom)

【题目描述】

智慧之神 *why* 发现了一个 N 行 M 列的矩阵。第一行的数字是 $1, 2, \dots, M$ 。第二行的数字是 $M + 1, M + 2, \dots, 2M$, 以此类推第 N 行的数字是 $(N - 1)M + 1, (N - 1)M + 2, \dots, NM$ 。

例如, $N = 3, M = 4$ 的矩阵是这样的:

```
1 2 3 4
5 6 7 8
9 10 11 12
```

对于身为智慧之神的 *why* 来说, 这个矩阵过于无趣。于是她决定改造这个矩阵, 改造会进行 K 次, 每次改造会将矩阵的某一行或某一列乘上一个数字, 你的任务是计算最终这个矩阵内所有数字的和, 答案对 $10^9 + 7$ 取模。

【输入格式】

第一行包含三个正整数 N, M, K , 表示矩阵的大小以及改造次数, 接下来的 K 行, 每行会是如下两种形式之一:

- 1. **R X Y**: 表示将矩阵的第 X 行变为原来的 Y 倍。
- 2. **S X Y**: 表示将矩阵的第 X 列变为原来的 Y 倍。

【输出格式】

输出一行一个整数, 表示最终矩阵内所有元素的和对 $10^9 + 7$ 取模的结果。

【输入输出样例1】

wisdom.in	wisdon.out
3 4 4 R 2 4 S 4 1 R 3 2 R 2 0	94

【输入输出样例1说明】

操作结束之后, 矩阵如下:

```
1 2 3 4
0 0 0 0
18 20 22 24
```

【输入输出样例2】

wisdom.in	wisdon.out
3 1 1 S 1 4	24

【输入输出样例3】

wisdom.in	wisdon.out
2 4 4 S 2 0 S 2 3 R 1 5 S 1 3	80

【数据规模与约定】

- 存在 50% 的测试数据保证 $1 \leq N, M \leq 10^3$ 。

对于所有的测试数据，保证 $1 \leq N, M \leq 10^6, 1 \leq K \leq 1000, 1 \leq X \leq N, 0 \leq Y \leq 10^9$ 。

矩阵入门 (matrix)

【题目描述】

Hikari 酱有一个 $n \times m$ 的矩阵，每个元素都是 0 或 1。众所周知，Hikari 很喜欢 1，但是她觉得这个矩阵的 1 太多了，她想进行若干次操作，每次操作可以取反任意一行或一列的所有元素。Hikari 想知道，如果她能进行任意次操作，矩阵中的 1 的个数最少是多少。

【输入格式】

第一行两个正整数 n, m 。

接下来 n 行，每行一个长度为 m 的 01 串，表示这个矩阵。

【输出格式】

输出一行一个整数，表示答案。

【输入输出样例1】

matrix.in	matrix.out
3 4 0110 1010 0111	2

【数据规模与约定】

对于 10% 的数据， $n, m \leq 4$;

对于 20% 的数据， $m \leq 5000$;

对于 40% 的数据， $n \leq 12$;

对于 70% 的数据， $n \leq 18$;

对于 100% 的数据， $n \leq 22, m \leq 2 \times 10^5$ 。

MEX 求和 (mex)

【题目描述】

对于一个非负整数序列 A ，定义 $MEX(A)$ 为最小的不在 A 中出现的非负整数。

现在给定非负整数序列 B_1, B_2, \dots, B_n ，求所有满足 $0 \leq A_i \leq B_i (i = 1, \dots, n)$ 的非负整数序列 A 的 $MEX(A)$ 之和，答案对 $10^9 + 7$ 取模。

【输入格式】

第一行：一个正整数 n 。

第二行： n 个整数 B_1, B_2, \dots, B_n 。

【输出格式】

一个整数，表示答案。

【输入输出样例1】

mex.in	mex.out
5 6 8 3 2 1	2824

【输入输出样例2】

mex.in	mex.out
8 3 7 1 11 0 4 1 9	208746

【数据规模与约定】

本题采用捆绑测试。对于所有的测试数据，保证 $1 \leq n \leq 5000, 0 \leq B_i \leq 10^9$ 。

子任务编号	$n \leq$	$B_i \leq$	分值
1	7	7	10
2	18	10^9	15
3	300	300	15
4	300	10^9	20
5	5000	5000	20
6	5000	10^9	20