

NOIP2024 模拟赛

GDFZ

时间：2024 年 10 月 11 日

题目名称	矩阵	比赛	街道	字符串
题目类型	传统题	传统题	传统题	传统题
目录	matrix	game	street	string
可执行文件名	matrix	game	street	string
输入文件名	matrix.in	game.in	street.in	string.in
输出文件名	matrix.out	game.out	street.out	string.out
每个测试点时限	1 秒	3 秒	2 秒	3 秒
内存限制	512 MB	512 MB	512 MB	1024 MB
子任务数目	10	20	5	3
测试点是否等分	是	是	否	否

提交源程序文件名

对于 C++ 语言	matrix.cpp	game.cpp	street.cpp	string.cpp
-----------	------------	----------	------------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

- 1. 测试机器：CPU(AMD Ryzen 5 3600 6-Core Processor *12), RAM 8.0G。
- 2. 系统环境：NOI Linux 2.0(基于 Ubuntu 20.04.1)。
- 3. 文件名（程序名和输入输出文件名）必须使用英文小写。
- 4. C/C++ 中函数 `main()` 返回类型必须是 `int`，程序正常结束返回值必须是 0。
- 5. 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）。
- 6. 选手提交的程序源文件必须不大于 100KB。
- 7. 程序可使用的栈空间内存限制与题目的内存限制一致。

矩阵 (matrix)

【题目描述】

给出两个 $n \times m$ 的矩阵 A 和 B , 你可以进行若干次操作。每次操作你可以使 A 或 B 的某一行或者某一列的所有元素增加 1。

问至少要多少次操作, 才能使 A 和 B 相等。

【输入格式】

从文件 *matrix.in* 中读入数据。

第一行一个正整数 T , 表示数据组数。

每组数据的第一行两个正整数 n, m 。

接下来 n 行, 每行 m 个非负整数 $A_{i,j}$, 表示矩阵 A 。

接下来 n 行, 每行 m 个非负整数 $B_{i,j}$, 表示矩阵 B 。

【输出格式】

输出到文件 *matrix.out* 中。

对于每组数据, 输出一行一个整数表示答案。如果无论怎么操作都不能使得 A 和 B 相等, 输出 -1 。

【样例 1 输入】

```
1 1
2 3 3
3 1 1 1
4 1 1 1
5 1 1 1
6 3 2 2
7 2 1 1
8 2 1 1
```

【样例 1 输出】

```
1 2
```

【样例 2】

见选手目录下的 *matrix/matrix2.in* 与 *matrix/matrix2.ans*。

【数据范围】

测试点编号	特殊限制
1	$n \times m \leq 5$
2 ~ 3	$n \times m \leq 100$
4 ~ 6	$n, m \leq 100$
7 ~ 10	$n \times m \leq 10^5, 1 \leq T \leq 5, 0 \leq A_{i,j}, B_{i,j} \leq 10^9$

比赛 (game)

【题目描述】

有一场比赛，分两天举行。这场比赛有 N 个选手参加，在两天比完赛之后，会根据每名选手两天成绩之和排序（如有同分则随机排序），然后给第一、二、三名分别颁发金、银、铜牌。

现在比赛已经进行完第一天了，你得到了这 N 个选手第一天的成绩，分别为 A_1, A_2, \dots, A_N ，满足它们均为整数且 $\forall 1 \leq i < j \leq N, A_i \neq A_j$ 。同时，作为第二天的出题人，你知道这些选手在第二天的得分将会是一个 $1, 2, \dots, N$ 的排列。

给定 P ，你想统计满足下述条件的三元组 (X, Y, Z) 的个数：

- 若 $P = 1$ ，则 $A_X > A_Y > A_Z$ ；否则 $A_X > A_Z > A_Y$ 。
- 第 X, Y, Z 名选手在第二天结束后可能分别获得金、银、铜牌。

【输入格式】

从文件 `game.in` 中读入数据。

每个测试点有多组询问。

第一行一个整数 T 表示询问组数。

接下来对于每组询问：

- 第一行两个整数表示 N, P 。
- 第二行 N 个整数表示 A_1, A_2, \dots, A_N 。

【输出格式】

输出到文件 `game.out` 中。

共 T 行，每行一个整数表示答案。

【样例 1 输入】

```
1 2
2 5 1
3 3 1 6 4 8
4 5 2
5 6 4 1 7 2
```

【样例 1 输出】

```
1 4
2 3
```

【样例 1 解释】

对于第一组询问，有 4 种可能的 (X, Y, Z) ：

- $(X, Y, Z) = (5, 3, 1)$ ，若第二天分数为 $(3, 4, 2, 1, 5)$ ，则总分为 $(6, 5, 8, 5, 13)$ ，满足条件；
- $(X, Y, Z) = (5, 3, 2)$ ，若第二天分数为 $(2, 4, 3, 1, 5)$ ，则总分为 $(5, 5, 9, 5, 13)$ ，满足条件；
- $(X, Y, Z) = (5, 3, 4)$ ，若第二天分数为 $(1, 2, 3, 4, 5)$ ，则总分为 $(4, 3, 9, 8, 13)$ ，满足条件；
- $(X, Y, Z) = (5, 4, 1)$ ，若第二天分数为 $(4, 2, 1, 3, 5)$ ，则总分为 $(7, 3, 7, 7, 13)$ ，满足条件。

对于第二组询问 3 种可能的 (X, Y, Z) 为 $(4, 2, 1), (4, 5, 1), (4, 5, 2)$ 。

【样例 2 输入】

```
1 2
2 10 1
3 6 4 11 14 3 17 13 18 8 10
4 10 2
5 18 14 19 4 12 1 7 15 9 5
```

【样例 2 输出】

```
1 26
2 14
```

【样例 3】

见选手目录下的 `game/game3.in` 与 `game/game3.ans`。

【数据范围】

对于所有数据，保证 $1 \leq T \leq 3, 1 \leq N \leq 10^6, P \in \{1, 2\}, a_i \leq 10^9$ 。

测试点编号	$P =$	$N \leq$
1	1	30
2		70
3		200
4		600
5		2000
6		8000
7		50000
8		2×10^5
9		5×10^5
10		10^6
11	2	30
12		70
13		200
14		600
15		2000
16		8000
17		50000
18		2×10^5
19		5×10^5
20		10^6

街道 (street)

【题目描述】

你住在一条街上，这条街上顺序排放了编号为 $0, 1, \dots, n$ 的 $n + 1$ 栋楼。从第 $i (0 \leq i \leq n)$ 栋楼出发，你可以花一分钟走到第 $i - 1$ 栋楼（若 $i - 1 \geq 0$ ）或第 $i + 1$ 栋楼（若 $i + 1 \leq n$ ）。

这条街上还有 K 个地铁站，分别为 $0 = S_0 < S_1 < \dots < S_{K-1} = n$ ，从第 $S_i (0 \leq i < K)$ 栋楼出发，你可以花一分钟做地铁到第 S_{i-1} 栋楼（若 $i - 1 \geq 0$ ）或第 S_{i+1} 栋楼（若 $i + 1 < K$ ）。

即使这样，你还是觉得交通太不方便了，你准备修一条公交线路。具体地，你可以建 L 个公交站（ L 是你选定的），分别为 $0 = T_0 < T_1 < \dots < T_{L-1} = n$ ，建好之后，从第 $T_i (0 \leq i < L)$ 栋楼出发，你可以花一分钟做公交到第 T_{i-1} 栋楼（若 $i - 1 \geq 0$ ）或第 T_{i+1} 栋楼（若 $i + 1 < L$ ）。

你修公交线路是为了让这条街上的交通更加便捷，所以你需要满足：

- 修好之后从第 0 栋楼出发，可以用不超过 X 分钟到这条街上任意一栋楼（忽略换乘时间）。
- 为了方便换乘，每栋楼若有地铁站，则必须要有公交站，即 $\{S_i\}_{i=0}^{K-1} \subset \{T_i\}_{i=0}^{L-1}$ 。

你要求出合法修建公交线路的条数，即满足条件的不同的集合 $\{T_i\}_{i=0}^{L-1}$ 的个数。答案对 $10^9 + 7$ 取模。

【输入格式】

从文件 *street.in* 中读入数据。

第一行三个整数 n, K, X ，意义见题目描述。

第二行 K 个整数 S_0, S_1, \dots, S_{K-1} ，意义见题目描述。

【输出格式】

输出到文件 *street.out* 中。

一行一个整数表示答案。

【样例 1 输入】

```
1 7 2 3
2 0 7
```

【样例 1 输出】

155

【样例 1 解释】

不合法的方案有 $\{0, 7\}, \{0, 1, 7\}, \{0, 1, 2, 7\}, \{0, 1, 6, 7\}, \{0, 1, 2, 6, 7\}, \{0, 1, 2, 3, 6, 7\}, \{0, 1, 2, 5, 6, 7\}, \{0, 1, 2, 3, 5, 6, 7\}, \{0, 1, 2, 3, 4, 5, 6, 7\}$, 共 9 种。于是合法的方案共 $2^6 - 9 = 55$ 种。

【样例 2】

见选手目录下的 `street/street2.in` 与 `street/street2.ans`。
该样例满足子任务 2 的限制。

【样例 3】

见选手目录下的 `street/street3.in` 与 `street/street3.ans`。
该样例满足子任务 4 的限制。

【数据范围】

对于所有测试点，满足 $2 \leq K \leq 2500, 1 \leq X \leq 2500, S_0 = 0, S_{K-1} = n, 1 \leq S_{i+1} - S_i \leq 10000$ 。

子任务编号	特殊性质	分值
1	$n, K, X \leq 15$	12
2	$K, X \leq 15, S_{i+1} - S_i \leq 15$	9
3	$K, X \leq 40, S_{i+1} - S_i \leq 40$	26
4	$K, X \leq 300, S_{i+1} - S_i \leq 300$	16
5	无特殊限制	37

字符串 (string)

【题目描述】

你有一个有小写字母组成的字符串 S ，你将它的所有非空子序列按字典序从小到大排序并去重，得到了一个字符串序列 T_1, T_2, \dots, T_m 。比如，若 $S = \text{aqua}$ ，则序列 $\{T\}_{i=1}^m$ 为 $[a, aa, aq, aqa, aqu, aqua, au, aua, q, qa, qu, qua, u, ua]$ 。

有 Q 组询问，每组询问给定 K, p ，你要求出 T_K 的最后 p 个字符，若 $K > m$ 输出 -1 ，若 T_K 不足 p 个字符则求出它的所有字符。

【输入格式】

- 从文件 `string.in` 中读入数据。
- 第一行一个有小写字母组成的字符串 S 。
- 第二行一个整数 Q ，表示询问组数。
- 接下来 Q 行每行两个整数 K, p ，表示询问。

【输出格式】

- 输出到文件 `string.out` 中。
- 共 Q 行，每行为 -1 或一个由小写字母组成的字符串，依次表示每组询问的答案。

【样例 1 输入】

```
1 aqua
2 16
3 1 10
4 2 10
5 3 10
6 4 10
7 5 10
8 6 10
9 7 10
10 8 10
11 9 10
12 10 10
13 11 10
```

```
14 12 10
15 13 10
16 14 10
17 15 10
18 16 10
```

【样例 1 输出】

```
1 a
2 aa
3 aq
4 aqa
5 aqu
6 aqua
7 au
8 aua
9 q
10 qa
11 qu
12 qua
13 u
14 ua
15 -1
16 -1
```

【样例 1 解释】

见题目描述。

【样例 2 输入】

```
1 tourist
2 2
3 76 10
4 76 4
```

【样例 2 输出】

```
1 tourist
2 rist
```

【样例 2 解释】

对于第二个询问， T_{76} 为 tourist，但只用输出后 4 个字符，即 rist。

【样例 3】

见选手目录下的 *string/string3.in* 与 *string/string3.ans*。

【数据范围】

对于所有测试点，满足 $1 \leq Q \leq 10^5, 1 \leq p, |S| \leq 3 \times 10^5, \sum p \leq 10^6, 1 \leq K \leq 10^{18}$ ， S 仅包含小写字母。

子任务编号	特殊性质	分值
1	$Q = 1$	15
2	$K \leq 10^6$	23
3	无特殊限制	62