

NOIP 2018

Day 2

dy0607

October 6, 2018

题目名称	Reverse	Silhouette	Seat
源文件名	reverse	silhouette	seat
输入文件名	reverse.in	silhouette.in	seat.in
输出文件名	reverse.out	silhouette.out	seat.out
题目类型	传统型	传统型	传统型
每个测试点时限	1.0s	1.0s	1.0s
空间限制	512MB	512MB	512MB
编译命令	-lm -O2 -std=c++11		

Notes:

1. 评测在Ubuntu16.04(64bit)上进行，评测时开启无限栈；
2. 评测机配置为Intel® Pentium(R) CPU G2030 @ 3.00GHz × 2，内存4GB；
3. 遇到原题请不要大喊“这不是xx上的xx题吗”，可以AK后提前离场；
4. 输入量较大，建议使用较快的读入方式。
5. 题目难度可能与顺序无关。

1 Reverse

1.1 Description

小G有一个长度为 n 的01串 T ，其中只有 $T_S = 1$ ，其余位置都是0。现在小G可以进行若干次以下操作：

- 选择一个长度为 K 的连续子串（ K 是给定的常数），翻转这个子串。

对于每个 $i, i \in [1, n]$ ，小G想知道最少要进行多少次操作使得 $T_i = 1$ 。特别的，有 m 个“禁止位置”，你需要保证在操作过程中1始终不在任何一个禁止位置上。

1.2 Input

从文件`reverse.in`中读入数据。

第一行四个整数 n, K, m, S 。

接下来一行 m 个整数表示禁止位置。

1.3 Output

输出到文件`reverse.out`中。

输出一行 n 个整数，对于第 i 个整数，如果可以通过若干次操作使得 $T_i = 1$ ，输出最小操作次数，否则输出 -1 。

1.4 Sample1

1.4.1 Input

6 2 0 1

1.4.2 Output

0 1 2 3 4 5

1.5 Sample2

1.5.1 Input

10 4 3 3

2 5 10

1.5.2 Output

2 -1 0 1 -1 1 2 3 2 -1

1.6 Sample3

见选手目录下的`reverse/reverse3.in`与`reverse/reverse3.ans`。

1.7 Sample4

见选手目录下的`reverse/reverse4.in`与`reverse/reverse4.ans`。

1.8 Subtasks

对于所有数据，有 $1 \leq n \leq 10^5, 1 \leq S, k \leq n, 0 \leq m \leq n$ 。

保证 S 不是禁止位置，但禁止位置可能有重复。

- Subtask1(24%), $n \leq 10$.
- Subtask2(22%), $n \leq 10^3$.
- Subtask3(3%), $k = 1$.
- Subtask4(8%), $k = 2$.
- Subtask5(43%), 没有特殊的约束。

2 Silhouette

2.1 Description

有一个 $n \times n$ 的网格，在每个格子上堆叠了一些边长为1的立方体。

现在给出这个三维几何体的正视图和左视图，求有多少种与之符合的堆叠立方体的方案。两种方案被认为是不同的，当且仅当某个格子上立方体的数量不同。

输出答案对 $10^9 + 7$ 取模的结果。

2.2 Input

从文件`silhouette.in`中读入数据。

第一行一个整数 n 。

第二行 n 个整数，第 i 个表示正视图中从左到右第 i 个位置的高度 A_i 。

第三行 n 个整数，第 i 个表示左视图中从左到右第 i 个位置的高度 B_i 。

2.3 Output

输出到文件`silhouette.out`中。

输出一行表示答案。

2.4 Sample1

2.4.1 Input

```
2
1 2
2 1
```

2.4.2 Output

```
5
```

2.4.3 Explanation

正视图和左视图：

```

  ┌─┐ ┌─┐
  │ │ │ │ │ │
  └─┘ └─┘

```

如果用 2×2 的矩阵来表示每个格子上堆叠的立方体个数，则五种方案可以表示为：

```
1 2   0 2   1 2   0 2   1 2
1 1   1 0   1 0   1 1   0 1
```

2.5 Sample2

2.5.1 Input

```
3
3 1 3
2 3 2
```

2.5.2 Output

```
175
```

2.6 Sample3

2.6.1 Input

```
3
1 1 1
3 3 3
```

2.6.2 Output

```
0
```

2.7 Sample4

见选手目录下的 *silhouette/silhouette4.in* 与 *silhouette/silhouette4.ans*.

2.8 Sample5

见选手目录下的 *silhouette/silhouette5.in* 与 *silhouette/silhouette5.ans*.

2.9 Subtasks

对于所有数据，有 $1 \leq n \leq 10^5, 1 \leq A_i, B_i \leq 10^9$.

- Subtask1(3%), $n = 1$.
- Subtask2(14%), $n \leq 3, A_i, B_i \leq 4$.

- Subtask3(17%), $n \leq 16$.
- Subtask4(24%), $n \leq 100$.
- Subtask5(13%), $n \leq 3000$.
- Subtask6(18%), A_i, B_i 分别构成了一个1至 n 的排列.
- Subtask7(11%), 没有特殊的约束.

3 Seat

3.1 Description

有 $n + 2$ 个座位等距地排成一排，从左到右编号为 0 至 $n + 1$ 。

最开始时 0 号以及 $n + 1$ 号座位上已经坐了一个小 G，接下来会有 n 个小 G 依次找一个空座位坐下。由于小 G 们坐得太近就容易互相博弈，每个小 G 会找一个当前离最近的小 G 距离最近的座位坐下。如果有多个备选座位，这个小 G 会等概率选择其中一个。

给出 n ，求第 i 个坐下的小 G 坐在 j 号座位的概率，对 P 取模。具体来说，如果答案化为最简分数可以表示为 $\frac{a}{b}$ ，你需要输出 $a \times b^{-1}$ ，其中 $b^{-1} = b^{P-2} \pmod{P}$ 。

3.2 Input

从文件 *seat.in* 中读入数据。

一行两个整数 n, P 。

3.3 Output

输出到文件 *seat.out* 中。

输出 n 个整数，第 i 行第 j 个整数表示第 i 个小 G 坐在第 j 个座位的概率。

3.4 Sample1

3.4.1 Input

4 10007

3.4.2 Output

0 5004 5004 0

3336 1668 1668 3336

3336 1668 1668 3336

3336 1668 1668 3336

3.4.3 Explanation

第一个小 G 会在中间两个位置中随机选择一个，接下来无论选哪个位置最近的距离都是 1。

$\frac{1}{2} = 5004 \pmod{10^4 + 7}$, $\frac{1}{3} = 3336 \pmod{10^4 + 7}$, $\frac{1}{6} = 1668 \pmod{10^4 + 7}$

3.5 Sample2

见选手目录下的 *seat/seat2.in* 与 *seat/seat2.ans*。

3.6 Subtasks

对于所有数据，满足 $2 \leq n \leq 1024$, $2000 \leq P \leq 30000$ ， P 是质数。

本题共25个测试点，每个测试点4分。

测试点编号	n
1	$= 1$
2	$= 2$
3, 4	≤ 10
5, 6	≤ 20
7, 8	≤ 32
9, 10	≤ 48
11, 12	≤ 64
13	≤ 80
14	≤ 100
15	$= 127$
16, 17, 18, 19, 20	≤ 300
21	$= 511$
22	$= 1023$
23, 24, 25	≤ 1024