

# 旋转子段

(rotate.pas/c/cpp/1000ms/512MB)

## 【问题描述】

ZYL有N张牌编号分别为1, 2, ……, N。他把这N张牌打乱排成一排, 然后他要做一次旋转使得旋转后固定点尽可能多。如果第i个位置的牌的编号为i, 我们就称之为固定点。旋转可以被认为是将其中的一个子段旋转180度, 这意味着子段的第一张牌和最后一张牌交换位置, 以及第二张牌和倒数第二张牌交换位置, 等等。写一个程序, 找到旋转子段(子段长度可以为1)。

## 【输入】

第一行包含一个整数 N ( $1 \leq N \leq 100\ 000$ )。

第二行有N个数, 第i个数表示旋转之前第i个位置的牌的编号。

## 【输出】

找到固定点最多的旋转所选的子段, 输出旋转之后固定点的个数。

## 【输入输出样例】

样例 1		样例 2	
rotate.in	rotate.out	rotate.in	rotate.out
4 3 2 1 4	4	2 1 2	2

### 样例解释:

在样例1中, 只需要旋转的子段[3, 2, 1], 将排列变成1 2 3 4, 旋转后所有的牌都为固定点。答案为4。

在样例2中, 所有的牌已经在固定点, 旋转子段[1]或者子段[2], 答案为2。

## 【数据范围】

30%的数据满足:  $N \leq 500$ ;

60%的数据满足:  $N \leq 5000$ ;

100%的数据满足:  $1 \leq N \leq 100\ 000$ 。

# 走格子

(cell.pas/c/cpp/1000ms/512MB)

## 【问题描述】

CYJ想找到他的小伙伴FPJ，.CYJ和FPJ现在位于一个房间里,这个房间的布置可以看成是一个N行M列的矩阵,矩阵内的每一个元素会是下列情况中的一种:

1. 障碍区域—这里有一堵墙(用‘#’表示).
2. 这是CYJ最开始的区域(用‘C’表示).
3. 这是FPJ在的区域(用‘F’表示).
4. 空区域(用‘.’表示).

CYJ携带了一个所谓传送枪的东西,这是一把可以创造传送门的枪械,在每一次行动中,他可以选择下列操作中的一项:

1. 移向一个相邻的格子中(上,下,左,右,不能移到墙在的格子里).这个操作要消耗一个单位的时间.
2. 转向一个墙(不需要相邻,只需面向即可),向其发射传送门,传送门会留在墙内面向你的地方(至多只能同时存在两扇传送门),若墙上已经有两扇传送门,而你发射了第三扇,那么最初发射的那一扇会消失.同时,你无法在一个位置制造两扇传送门(这个操作不会耗费时间)。
3. 如果他与一块墙壁相邻且面前有一扇传送门,那么他可以移动到另一扇传送门前方的格子.这个操作会耗费一个单位的时间.

CYJ想要知道自己最少需要多少时间才能够从起点(‘C’)到达终点(‘F’).

**请注意:**我们保证地图边缘会是一圈墙壁且一定存在‘C’,‘F’.

## 【输入】

第一行输入两个正整数  $N$  和  $M$  ,( $4 \leq N, M \leq 500$ ).表示地图大小。

接下来的N行每行一个长度为M的字符串.表示地形。

## 【输出】

你需要输出最少的到达终点的时间,如果不能到达请输出"no"。

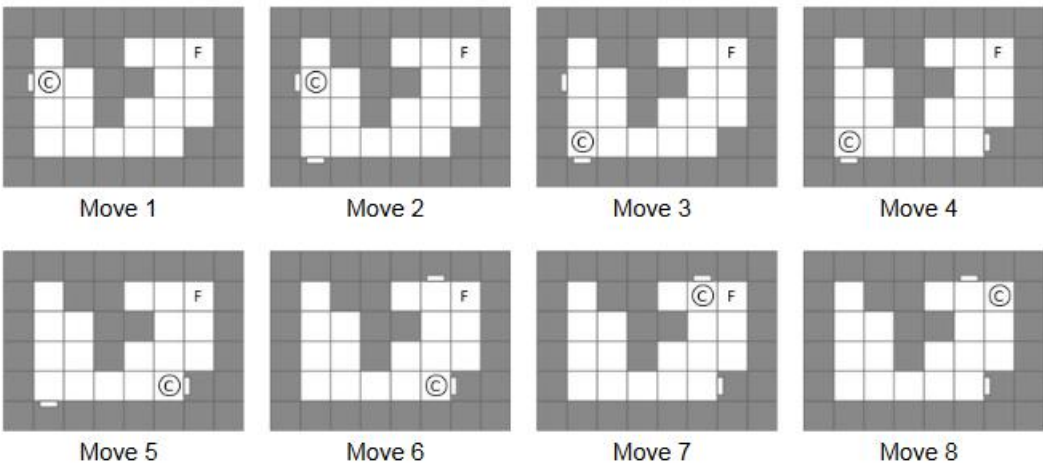
## 【输入输出样例】

样例 1	样例 2	样例 3
<b>cell.in</b>	<b>cell.in</b>	<b>cell.in</b>
4 4	6 8	#####
####	#####	#C#.#
#.F#	###..F#	###F#
#C.#	#C.##..#	#####
####	##.#...#	
<b>cell.out</b>	#.....##	cell.out

2	#####  cell.out  4	no
---	--------------------------------	----

**样例 2 解释：**

从 C 点(3,2)开始,我们首先向左发射传送门,再向下发射传送门,向左进入传送门,到达(5,2),向右发射传送门,向下进入传送门,到达(5,6),向上发射传送门,向右进入传送门,到达(2,6),向右移动,到达 F.



**【数据范围】**

- 50%的数据满足：  $4 \leq N, M \leq 15$ ;
- 100%的数据满足：  $4 \leq N, M \leq 500$ ;



## 柱状图

(column. pas/c/cpp/1500ms/512MB)

### 【问题描述】

WTH获得了一个柱状图,这个柱状图一共有 $N$ 个柱子,最开始第 $i$ 根柱子的高度为 $x_i$ ,他现在要将这个柱状图排成一个屋顶的形状,屋顶的定义如下:

1. 屋顶存在一个最高的柱子,假设为 $i$ ,最终高度为 $h_i$ .它是所有柱子之中最高的.
2. 第 $j$ 根柱子的高度为 $h_j=h_i-|i-j|$ ,但这个高度必须大于0,否则就是不合法的.

WTH可以对一个柱子做的操作只有将其高度加一或减一, WTH正忙着享受自己的人赢生活于是他将把这个柱状图变成屋顶的任务交给你.你需要求出最少进行多少次操作才能够把这个柱状图变成一个屋顶形状.

### 【输入】

第一行包含一个正整数  $N(1 \leq N \leq 100\,000)$ .

第二行包含  $N$  个用空格隔开的正整数,表示  $x_i$ ,含义如题面。

### 【输出】

输出最少进行多少个操作才能够把这个柱状图变成屋顶的形状。

### 【输入输出样例】

样例 1		样例 2	
column.in	column.out	column.in	column.out
4 1 1 2 3	3	5 4 5 7 2 2	4

**样例的解释:** 例一升高2, 3, 4号柱子一个单位高度是操作最少的方法之一,最高处为第四个柱子。例二降低第三根柱子三个高度,升高第四个柱子一个高度。最高处为第2个柱子。

### 【数据范围】

30%的数据满足:  $1 \leq N \leq 100$ 。

60%的数据满足:  $1 \leq N \leq 5000$ 。

100%的数据满足:  $1 \leq N \leq 100000$ ,  $1 \leq h_i \leq 10^9$ 。