Leaderboard

Discussions

Editorial

**Function Description**

Complete the function *minimumBribes* in the editor below.

minimumBribes has the following parameter(s):

- *int q[n]*: the positions of the people after all bribes

**Returns**

- No value is returned. Print the minimum number of bribes necessary or `Too chaotic` if someone has bribed more than $2$ people.

**Input Format**

The first line contains an integer $t$, the number of test cases.

Each of the next $t$ pairs of lines are as follows:
- The first line contains an integer $t$, the number of people in the queue
- The second line has $n$ space-separated integers describing the final state of the queue.

**Constraints**

- $1 \leq t \leq 10$
- $1 \leq n \leq 10^5$

**Subtasks**

For $60\%$ score $1 \leq n \leq 10^3$
For $100\%$ score $1 \leq n \leq 10^5$

**Sample Input**

```
STDIN       Function
-----       --------
2           t = 2
5           n = 5
2 1 5 3 4   q = [2, 1, 5, 3, 4]
5           n = 5
2 5 1 3 4   q = [2, 5, 1, 3, 4]
```

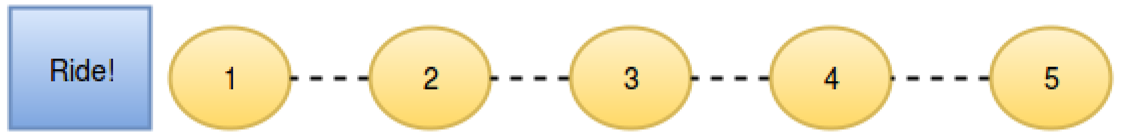
**Sample Output**

```
3
Too chaotic
```

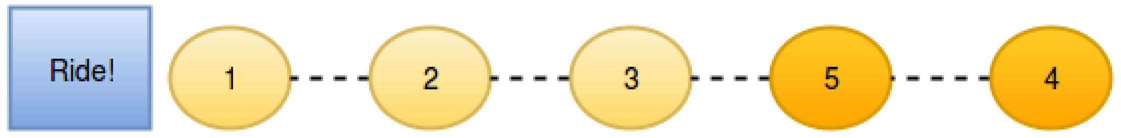
**Explanation**

**Test Case 1**

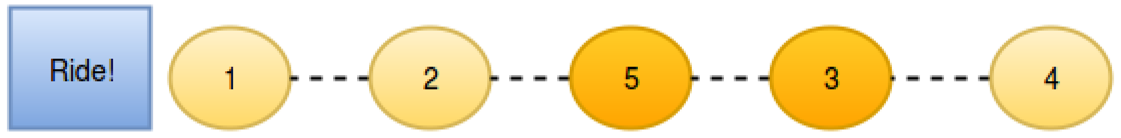
The initial state:



After person $5$ moves one position ahead by bribing person $4$:



Now person $5$ moves another position ahead by bribing person $3$:



And person $2$ moves one position ahead by bribing person $1$:

```
# Complete the 'minimumBribes' function below.
#
# The function accepts INTEGER_ARRAY q as parameter.
#

def minimumBribes(q):
    # using bubble sort, get the array back to it's initial state
    # number of swaps will be = to number of bribes
    bribes = 0

    # keep track of individual element's number of swaps
    # if it exceeded 2, return 'Too chaotic'
    bribes_map = {}

    for i in range(len(q)):
        swaps = True
        for j in range(len(q)-1):
            if q[j] > q[j+1]:
                bribes_map[q[j]] = bribes_map.get(q[j],0)+1

                if bribes_map[q[j]] > 2:
                    print('Too chaotic')
                    return

                q[j], q[j+1] = q[j+1], q[j]
                bribes +=1

                swaps = False

        # optimization
        if swaps is True:
            print(bribes)
            return

    print(bribes)
    return
```

Line: 19 Col: 1

Upload Code as File    □ Test against custom input    Run Code    Submit Code

## Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- ✓ **Test case 0**
- ✓ Test case 1
- ✓ Test case 2
- ✓ Test case 3
- ✓ Test case 4
- ✓ Test case 5
- ✓ Test case 6

Compiler Message

```
Success
```

Input (stdin)    Download

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

Expected Output    Download