

2020 Canadian Computing Olympiad  
Day 1, Problem 2  
**Exercise Deadlines**

**Time Limit: 1 second**

**Problem Description**

Bob has  $N$  programming exercises that he needs to complete before their deadlines. Exercise  $i$  only takes one time unit to complete, but has a deadline  $d_i$  ( $1 \leq d_i \leq N$ ) time units from now.

Bob will solve the exercises in an order described by a sequence  $a_1, a_2, \dots, a_N$ , such that  $a_1$  is the first exercise he solves,  $a_2$  is the second exercise he solves, and so on. Bob's original plan is described by the sequence  $1, 2, \dots, N$ . With one *swap* operation, Bob can exchange two adjacent numbers in this sequence. What is the minimum number of swaps required to change this sequence into one that completes all exercises on time?

**Input Specification**

The first line consists of a single integer  $N$  ( $1 \leq N \leq 200\,000$ ). The next line contains  $N$  space-separated integers  $d_1, d_2, \dots, d_N$  ( $1 \leq d_i \leq N$ ).

For 17 of the 25 marks available,  $N \leq 5000$ .

**Output Specification**

Output a single integer, the minimum number of swaps required for Bob to solve all exercises on time, or  $-1$  if this is impossible.

**Sample Input 1**

```
4
4 4 3 2
```

**Output for Sample Input 1**

```
3
```

**Explanation of Output for Sample Input 1**

One valid sequence is  $(1, 4, 3, 2)$ , which can be obtained from  $(1, 2, 3, 4)$  by three swaps.

**Sample Input 2**

3

1 1 3

**Output for Sample Input 2**

-1

**Explanation of Output for Sample Input 2**

There are two exercises that are due at time 1, but only one exercise can be solved by this time.