

F. Gardener Alex

time limit per test: 2 seconds
 memory limit per test: 256 megabytes
 input: standard input
 output: standard output

Gardener Alex loves to grow trees. We remind that tree is a connected acyclic graph on n vertices.

Today he decided to grow a rooted binary tree. A binary tree is a tree where any vertex has no more than two sons. Luckily, Alex has a permutation of numbers from 1 to n which he was presented at his last birthday, so he decided to grow a tree according to this permutation. To do so he does the following process: he finds a minimum element and makes it a root of the tree. After that permutation is divided into two parts: everything that is to the left of the minimum element, and everything that is to the right. The minimum element on the left part becomes the left son of the root, and the minimum element on the right part becomes the right son of the root. After that, this process is repeated recursively on both parts.

Now Alex wants to grow a forest of trees: one tree for each cyclic shift of the permutation. He is interested in what cyclic shift gives the tree of minimum depth. Unfortunately, growing a forest is a hard and long process, but Alex wants the answer right now. Will you help him?

We remind that cyclic shift of permutation $a_1, a_2, \dots, a_k, \dots, a_n$ for k elements to the left is the permutation $a_{k+1}, a_{k+2}, \dots, a_n, a_1, a_2, \dots, a_k$.

Input

First line contains an integer number n ($1 \leq n \leq 200\,000$) — length of the permutation.

Second line contains n integer numbers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$), and it is guaranteed that all numbers occur exactly one time.

Output

Print two numbers separated with space: minimum possible depth of a tree and how many elements we need to shift left to achieve this depth. The number of elements should be a number from 0 to $n - 1$. If there are several possible answers, print any of them.

Example

input	Copy
4 1 2 3 4	
output	Copy
3 2	

Note

The following picture depicts all possible trees for sample test and cyclic shifts on which they are achieved.

Codeforces Round #586 (Div. 1 + Div. 2)**Finished**

Practice

→ **Virtual participation**

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)→ **Practice**

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ **Clone Contest to Mashup**

You can clone this contest to a mashup.

[Clone Contest](#)→ **Submit?**

Language: GNU G++11 5.1.0

 Choose file: [选择文件](#) 未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

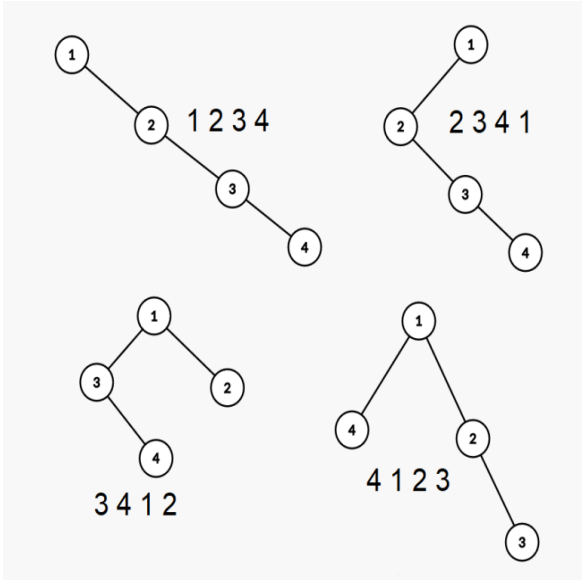
[Submit](#)→ **Problem tags**
[binary search](#) [data structures](#) [*2600](#)

No tag edit access

→ **Contest materials**

- Announcement #1 (en) 

- Announcement #2 (ru) ✕



[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Sep/19/2019 14:57:24^{UTC+8} (h2).
Desktop version, switch to [mobile version](#).
[Privacy Policy](#)

Supported by

