

Larry's Array

Larry has been given a permutation of a sequence of natural numbers incrementing from 1 as an array. He must determine whether the array can be sorted using the following operation any number of times:

- Choose any 3 consecutive indices and rotate their elements in such a way that $ABC \rightarrow BCA \rightarrow CAB \rightarrow ABC$.

For example, if $A = \{1, 6, 5, 2, 4, 3\}$:

```

A          rotate
[1,6,5,2,4,3] [6,5,2]
[1,5,2,6,4,3] [5,2,6]
[1,2,6,5,4,3] [5,4,3]
[1,2,6,3,5,4] [6,3,5]
[1,2,3,5,6,4] [5,6,4]
[1,2,3,4,5,6]

```

YES

On a new line for each test case, print YES if A can be fully sorted. Otherwise, print NO.

Function Description

Complete the *larrysArray* function in the editor below. It must return a string, either YES or NO.

larrysArray has the following parameter(s):

- A : an array of integers

Input Format

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

The next t pairs of lines are as follows:

- The first line contains an integer n , the length of A .
- The next line contains n space-separated integers $A[i]$.

Constraints

- $1 \leq t \leq 10$
- $3 \leq n \leq 1000$
- $1 \leq A[i] \leq n$
- $A_{sorted} =$ integers that increment by 1 from 1 to n

Output Format

For each test case, print YES if A can be fully sorted. Otherwise, print NO.

Sample Input

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

Sample Output

```
YES
YES
NO
```

Explanation

In the explanation below, the subscript of A denotes the number of operations performed.

Test Case 0:

$$A_0 = \{3, 1, 2\} \rightarrow \text{rotate}(3, 1, 2) \rightarrow A_1 = \{1, 2, 3\}$$

A is now sorted, so we print **yes** on a new line.

Test Case 1:

$$A_0 = \{1, 3, 4, 2\} \rightarrow \text{rotate}(3, 4, 2) \rightarrow A_1 = \{1, 4, 2, 3\}.$$

$$A_1 = \{1, 4, 2, 3\} \rightarrow \text{rotate}(4, 2, 3) \rightarrow A_2 = \{1, 2, 3, 4\}.$$

A is now sorted, so we print **yes** on a new line.

Test Case 2:

No sequence of rotations will result in a sorted A . Thus, we print **no** on a new line.