Larry's Array

Larry has a permutation of N numbers, A, whose unique elements range from 1 to N (i.e.: $A = \{a_1, a_2, \ldots, a_{N-1}, a_{N}\}$). He wants A to be sorted, so he delegates the task of doing so to his robot. The robot can perform the following operation as many times as it wants:

- Choose any 3 consecutive indices and rotate their elements in such a way that $A B C$ rotates to $B C A$, which rotates to $C A B$, which rotates back to $A B C$.

For example: if $A = \{1,6,5,2,4,3\}$ and the robot rotates $(6,5,2)$, A becomes $\{1,\textbf{5, 2, 6},4,3\}$.

On a new line for each test case, print $\scriptsize{\texttt{YES}}$ if the robot can fully sort A; otherwise, print $\scriptsize{\texttt{NO}}$.

Input Format

The first line contains an integer, T, the number of test cases.
The $2T$ subsequent lines each describe a test case over 2 lines:

1. An integer, N, denoting the size of A.
2. N space-separated integers describing A, where the i^{th} value describes element a_i.

Constraints

- $1 \le T \le 10$
- $3 \le N \le 1000$
- $1 \le a_i \le N \text{, where every element } a_i \text{ is unique.}$

Output Format

On a new line for each test case, print $\scriptsize{\texttt{YES}}$ if the robot can fully sort A; otherwise, print $\scriptsize{\texttt{NO}}$.

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 $\scriptsize{\texttt{YES}}$ 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 $\scriptsize{\texttt{YES}}$ on a new line.

Test Case 2:
No sequence of rotations will result in a sorted A. Thus, we print $\scriptsize{\texttt{NO}}$ on a new line.