

Piling Up! ★

Problem | Submissions | Leaderboard | Editorial

There is a horizontal row of n cubes. The length of each cube is given. You need to create a new vertical pile of cubes. The new pile should follow these directions: if $cube[i]$ is on top of $cube[j]$ then $sideLength[j] \geq sideLength[i]$.

When stacking the cubes, you can only pick up either the leftmost or the rightmost cube each time. Print Yes if it is possible to stack the cubes. Otherwise, print No.

Example

$blocks = [1, 2, 3, 8, 7]$

Result: No

After choosing the rightmost element, 7, choose the leftmost element, 1. After than, the choices are 2 and 8. These are both larger than the top block of size 1.

$blocks = [1, 2, 3, 7, 8]$

Result: Yes

Choose blocks from right to left in order to successfully stack the blocks.

Input Format

The first line contains a single integer T , the number of test cases.

For each test case, there are 2 lines.

The first line of each test case contains n , the number of cubes.

The second line contains n space separated integers, denoting the sideLengths of each cube in that order.

Constraints

$1 \leq T \leq 5$

$1 \leq n \leq 10^5$

$1 \leq sideLength < 2^{31}$

Output Format

For each test case, output a single line containing either Yes or No.

Sample Input

STDIN	Function
-----	-----
2	T = 2
6	blocks[] size n = 6
4 3 2 1 3 4	blocks = [4, 3, 2, 1, 3, 4]
3	blocks[] size n = 3
1 3 2	blocks = [1, 3, 2]

Sample Output

Yes
No

Explanation

In the first test case, pick in this order: left - 4, right - 4, left - 3, right - 3, left - 2, right - 1.

In the second test case, no order gives an appropriate arrangement of vertical cubes. 3 will always come after either 1 or 2.

1 # Enter your code here. Read input from STDIN. Print output to STDOUT