Your Piling Up! submission got 50.00 points. Share Tweet

Try the Next Challenge | Try a Random Challenge

Piling Up!

by **sandyeep**

Problem Submissions Leaderboard Discussions 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"*. Do not print the quotation marks.

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"* without the quotes.

Sample Input

```
2
6
4 3 2 1 3 4
3
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.

Submissions: 7021

Max Score: 50

Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

More

Current Buffer (saved locally, editable)

Python 3

```python
from collections import deque

def dq_pop_max(dq):
    try:
        if dq[0]>=dq[-1]:
            return dq.popleft()
        else:
            return dq.pop()
    except IndexError:
        return None

def check_stack_dq(dq):
    cur_value = pop_value = dq_pop_max(dq)
    while(pop_value):
        if cur_value>=pop_value:
            cur_value, pop_value = pop_value,dq_pop_max(dq)
        else:
            return 'No'
    return 'Yes'

testcases = int(input().strip())
dq_list = []
for i in range(testcases):
    num = int(input().strip())
    dq_list.append(deque([int(x) for x in input().strip().split(' ')]))

for dq in dq_list:
    print(check_stack_dq(dq))
```

Line: 25 Col: 72

Upload Code as File □ Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

Challenge your friends:

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

✓ Test Case #3 ✓ Test Case #4

You've earned 50.00 points. Next Challenge

Contest Calendar|Blog|Scoring|Environment|FAQ|About Us|Support|Careers|Terms Of Service|Privacy Policy|Request a Feature