CS1020E | Lab 9 | Exercise 1

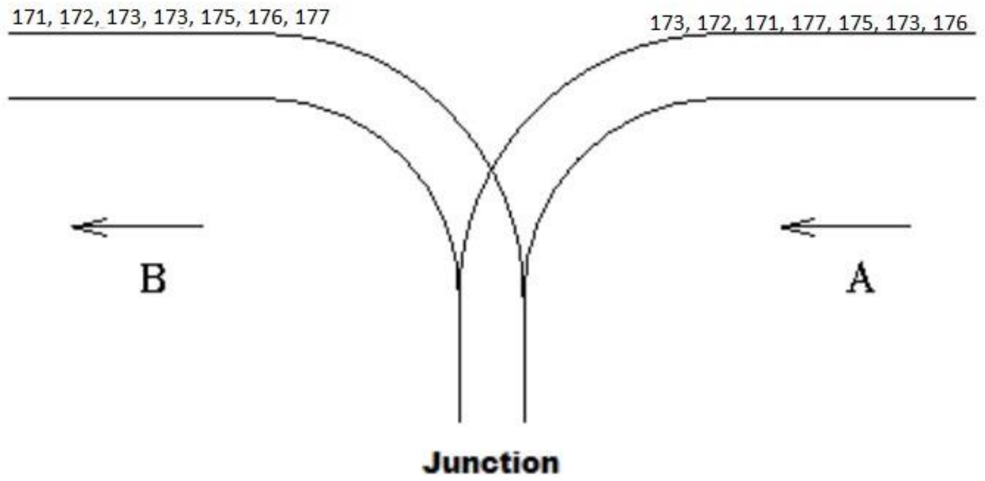
Stack Sort

Objectives

The objective of this exercise is to learn to use a **stack** to solve a problem.

Problem Description

A group of soldiers is in the middle of an espionage mission. In order to infiltrate the enemy's building, they went through a secret sewer tunnel. All of a sudden, an enemy who was patrolling inside the sewer found them. A contact happened but the soldiers managed to shoot the enemy down. Unfortunately, some of them are injured and they need to rearrange their formation so that they can better react to enemy attack. The tunnel is so narrow that the soldiers cannot cross one another. Luckily they found a junction so that they can rearrange their formation (see map below).



The soldiers are moving from Tunnel A and continuing in Tunnel B. they want to rearrange themselves so that their heights are in non-decreasing order from front to back.

Each solider can only move

1. from Tunnel A to the junction;
2. from the junction to Tunnel B.

He cannot dirrectly go from A to B, or from the junction to A, or from B to the junction.

Write a program to help John, the commander of the soldiers, to decide whether it is possible for them to make such a rearrangement.

Add your code only to the parts of the file indicated. Do not modify any other part of the given code, and do not add new files.

Inputs

The first line contains an integer *T*, which is the number of test cases. This is followed by *T* test cases. The first line of each test case contains an integer *N*, which is the number of soldiers. The second line contains *N* integers, which are the heights of the soldiers from front to back in Tunnel A.

Outputs

For each test case, output a line that contains YES if it is possible for the soldiers to rearrange themselves in non- decreasing order of height; otherwise output NO.

Sample Input

```
3
7
173 172 171 177 175 173 176
5
171 172 175 173 172
5
171 173 175 173 172
```

Sample Output

```
YES
YES
NO
```

Hint

Use a stack to simulate the movements into and out of the junction. At any time, the heights of the soldiers in the junction must be in non-decreasing order (starting from the top of the stack).

Additional Requirement

An efficient program is required. Specifically, the time complexity for each test case should be O(*N*). You would get **at most 50% of the marks** if your program's time complexity is worse than O(*N*).

Submission

You need to submit your completed **`StackSort.cpp`** to CodeCrunch (https://codecrunch.comp.nus.edu.sg/) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.