B. Array Fix

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an integer array a of length n.

You can perform the following operation any number of times (possibly zero): take any element of the array a, which is at least 10, delete it, and instead insert the digits that element consisted of in the same position, in order they appear in that element.

For example:

- if we apply this operation to the 3-rd element of the array $[12, 3, 45, 67]$, then the array becomes $[12, 3, 4, 5, 67]$.
- if we apply this operation to the 2-nd element of the array $[2, 10]$, then the array becomes $[2, 1, 0]$.

Your task is to determine whether it is possible to make a sorted in non-descending order using the aforementioned operation **any number of times (possibly zero)**. In other words, you have to determine if it is possible to transform the array a in such a way that $a_1 \le a_2 \le \cdots \le a_k$, where k is the current length of the array a.

Input

The first line contains a single integer t ($1 \le t \le 10^3$) — the number of test cases.

Each test case consists of two lines:

- the first line contains a single integer n ($2 \le n \le 50$).
- the second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 99$).

Output

For each test case, print `YES` if it is possible to make a sorted in non-decreasing order using the aforementioned operation; otherwise, print `NO`.

You can print each letter in any case. For example, `yes`, `Yes`, `YeS` will all be recognized as a positive answer.

Example

input Copy

```
3
4
12 3 45 67
3
12 28 5
2
0 0
```

output Copy

```
YES
NO
YES
```

Note

In the first example, you can split the first element, then the array becomes $[1, 2, 3, 45, 67]$.

In the second example, there is no way to get a sorted array.

In the third example, the array is already sorted.
