

1. Usernames Changes

A company has released a new internal system, and each employee has been assigned a username. Employees are allowed to change their usernames but only in a limited way. More specifically, they can choose letters at two different positions and swap them. For example, the username "bigfish" can be changed to "gibfish" (swapping 'b' and 'g') or "bighisf" (swapping 'f' and 'h'). The manager would like to know which employees can update their usernames so that the new username is smaller in alphabetical order than the original username.

For each username given, return either "YES" or "NO" based on whether that username can be changed (with one swap) to a new one that is smaller in alphabetical order.

Note: For two different strings A and B of the same length, A is smaller than B in alphabetical order when on the first position where A and B differ, A has a smaller letter in alphabetical order than B has.

For example, let's say `usernames = ["bee", "superhero", "ace"]`. For the first username, "bee", it is not possible to make one swap to change it to a smaller one in alphabetical order, so the answer is "NO". For the second username, "superhero", it *is* possible get a new username that is smaller in alphabetical order (for example, by swapping letters 's' and 'h' to get "hupersero"), so the answer is "YES". Finally, for the last username "ace", it is not possible to make one swap to change it to a smaller one in alphabetical order, so the answer is "NO". Therefore you would return the array of strings ["NO", "YES", "NO"].

Function Description

Complete the function `possibleChanges` in the editor below.

`possibleChanges` has the following parameter(s):

string `usernames[n]`: an array of strings denoting the usernames of the employees

Returns:

string[n]: an array of strings containing either "YES" or "NO" based on whether the i^{th} username can be changed with one swap to a new one that is smaller in alphabetical order

Constraints

- $1 \leq n \leq 10^5$
- The sum of lengths of all usernames does not exceed 10^6 .
- `usernames[i]` consists of only lowercase English letters.

▼ Input Format For Custom Testing

The first line of input contains an integer, n , denoting the number of employees.

Each line i of the n subsequent lines (where $0 \leq i < n$) contains a string, `usernames[i]`, denoting the username of the i^{th} employee.

▼ Sample Case 0

Sample Input For Custom Testing

```
1
hydra
```

Sample Output

```
YES
```

Explanation

There is just one username to consider in this case, and it is the username "hydra". One can swap, for example, the last two letters of it to get the new username "hydar". This is smaller in alphabetical order than "hydra", so the answer for this username is "YES" (without quotes).

▼ Sample Case 1

Sample Input For Custom Testing

```
3
foo
bar
baz
```

Sample Output

```
NO
YES
YES
```

Explanation

There are three usernames to consider in this case. For the first of them, "foo", it is not possible to make one swap to change it to a smaller one in alphabetical order, so the answer is "NO" (without quotes). For the second one, "bar", one can swap the letters "b" and "a" to get the new username "abr", which is smaller in alphabetical order, so the answer is "YES" (without quotes). Similarly, For the third username, "baz", one can swap the letters "b" and "a" to get the new username "abz", which is smaller in alphabetical order, so the answer is "YES" (without quotes).