

922 · Group Shifted Strings

Description

Given a string, we can "shift" each of its letter to its successive letter, for example: "abc" -> "bcd".

We can keep "shifting" which forms the sequence:

```
"abc" -> "bcd" -> ... -> "xyz"
```

Given a list of strings which contains only lowercase alphabets, group all strings that belong to the same shifting sequence.

You don't need to care about the order of the result.

Example

Example 1:

```
input: ["abc", "bcd", "acef", "xyz", "az", "ba", "a", "z"]
output: [["a", "z"], ["abc", "bcd", "xyz"], ["acef"], ["az", "ba"]]
```

Example 2:

```
input: ["a"]
output: [["a"]]
```

Tags

Company

Uber Google

```
import collections
class Solution:
    """
    @param strings: a string array
    @return: return a list of string array
    """
    def groupStrings(self, arr):
        # write your code her
        ans = collections.defaultdict(list)
        for ele in arr:
```

```
        temp = self.getAsciiCode(ele)
        ans[temp].append(ele)
    return list(ans.values())
def getAsciiCode(self, string):
    if len(string) == 1:
        return 1
    ans = ""
    for i in range(1, len(string)):
        val = ord(string[i]) - ord(string[i - 1])
        if val < 0:
            val = val + 26
        ans += str(val) + "#"
    return ans
```