Codewriting **300**

A string is said to be beautiful if each letter in the **string** appears at most as many times as **the previous letter in the alphabet within the string**; ie: `b` occurs no more times than `a`; `c` occurs no more times than `b`; etc. Note that letter `a` has no previous letter.

Given a string, check whether it is *beautiful*.

Example

- For `inputString = "bbbaacdafe"`, the output should be `solution(inputString) = true`.

  This string contains 3 `a`s, 3 `b`s, 1 `c`, 1 `d`, 1 `e`, and 1 `f` (and 0 of every other letter), so since there aren't any letters that appear more frequently than the previous letter, this string qualifies as beautiful.

- For `inputString = "aabbb"`, the output should be `solution(inputString) = false`.

  Since there are more `b`s than `a`s, this string is not beautiful.

- For `inputString = "bbc"`, the output should be `solution(inputString) = false`.

  Although there are more `b`s than `c`s, this string is not beautiful because there are no `a`s, so therefore there are more `b`s than `a`s.

Input/Output

- **[execution time limit] 4 seconds (py3)**

- **[input] string inputString**

  A string of lowercase English letters.

  Guaranteed constraints:
  `3 ≤ inputString.length ≤ 50`.

- **[output] boolean**

  Return `true` if the string is beautiful, `false` otherwise.

[Python 3] Syntax Tips

```
# Prints help message to the console
# Returns a string
```

```
def helloWorld(name):
    print("This prints to the console when you Run Tests")
    return "Hello, " + name
```