Easy Codewriting 2000

Given an array `strings`, determine whether it follows the sequence given in the `patterns` array. In other words, there should be no `i` and `j` for which `strings[i] = strings[j]` and `patterns[i] ≠ patterns[j]` or for which `strings[i] ≠ strings[j]` and `patterns[i] = patterns[j]`.

Example

- For `strings = ["cat", "dog", "dog"]` and `patterns = ["a", "b", "b"]`, the output should be `areFollowingPatterns(strings, patterns) = true`;
- For `strings = ["cat", "dog", "doggy"]` and `patterns = ["a", "b", "b"]`, the output should be `areFollowingPatterns(strings, patterns) = false`.

Input/Output

- **[execution time limit] 20 seconds (scala)**
- **[input] array.string strings**

  An array of strings, each containing only lowercase English letters.

  Guaranteed constraints:
  `1 ≤ strings.length ≤ 10^5`,
  `1 ≤ strings[i].length ≤ 10`.

- **[input] array.string patterns**

  An array of pattern strings, each containing only lowercase English letters.

  Guaranteed constraints:
  `patterns.length = strings.length`,
  `1 ≤ patterns[i].length ≤ 10`.

- **[output] boolean**

  Return `true` if `strings` follows `patterns` and `false` otherwise.

[Scala] Syntax Tips

```
def helloWorld(name: String): String = {
    println("This prints to the console when you Run Tests")
    "Hello, " + name
}
```