Easy Codewriting 1000

Given an array `a` that contains only numbers in the range from `1` to `a.length`, find the first duplicate **number** for which the second occurrence has the minimal index. In other words, if there are more than 1 duplicated numbers, return the **number** for which the second occurrence has a smaller index than the second occurrence of the other number does. If there are no such elements, return `-1`.

Example

- For `a = [2, 1, 3, 5, 3, 2]`, the output should be `firstDuplicate(a) = 3`.

  There are `2` duplicates: numbers `2` and `3`. The second occurrence of `3` has a smaller index than the second occurrence of `2` does, so the answer is `3`.

- For `a = [2, 2]`, the output should be `firstDuplicate(a) = 2`;

- For `a = [2, 4, 3, 5, 1]`, the output should be `firstDuplicate(a) = -1`.

Input/Output

- **[execution time limit] 20 seconds (scala)**

- **[input] array.integer a**

  Guaranteed constraints:

  `1 ≤ a.length ≤ 10^5`,

  `1 ≤ a[i] ≤ a.length`.

- **[output] integer**

  The element in `a` that occurs in the array more than once and has the minimal index for its second occurrence. If there are no such elements, return `-1`.

[Scala] Syntax Tips

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