Longest Continuous Increasing Subsequence

Problem condition:

Given an unsorted array of integers nums, return *the length of the longest* ***continuous increasing subsequence*** *(i.e. subarray)*. The subsequence must be **strictly** increasing.

Case: nums = [1, 2, 1, 2, 3, 4, 1, 2, 3]

Answer: 4

Algorithm:

At first, we need to ckeck if the nums array is not empty. If it is, then we need to return 0

If the array is not empty, we need to initialize two variables:
maxLength - the lentgh of the longest continuous increasing subsequence that we ever met during algorithm
currentLength - the length of the subsequence that we are currently considering

At first both of variables equals to 1 (Minimum possible length)

Main idea is to compare the current element of the array with the next element of the array.

If the next element is **larger** than the current one, then **increase currentLength** by one.

If the next element is **less** than or **equal** to the current one, then we need to find the **larger value** between **maxLentgh** and **currentLength** and white that value **to the maxLength variable, and currentLength is reset to 1**, because a new subsequence has started.

Repeat this algorithm till the end of the array and return the maximum between **maxLentgh** and **currentLength**

Example [1, 2, 1, 2, 3, 4, 1, 2, 3]:

At first we need to compare [1, 2], then [2, 1], then [1, 2], then [2, 3], then [3, 4], then [4, 1] . . . then [2, 3]

1 and 2 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

2 more than 1, so current increasing subsequence is **[1, 2]** and **currentLength = 2**

2 and 1 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

1 less than 2, so we need to find the maximum between maxLenght (1) and currentLength (2), maximum value is 2, so **at the moment** the longest increasing subsequence is **[1, 2]**, **maxLength = 2**, reset currentLength, **currentLength = 1**

1 and 2 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

2 more than 1, so current increasing subsequence is **[1, 2]** and **currentLength = 2**

1 and 2 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

3 more than 2, so current increasing subsequence is **[1, 2, 3]** and **currentLength = 3**

1 and 2 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

4 more than 3, so current increasing subsequence is **[1, 2, 3, 4]** and **currentLength = 4**

4 and 1 [1, 2, 1, 2, 3, 4, 1, 2, 3]**:**

1 less than 4, so we need to find the maximum between maxLenght (2) and currentLength (4), maximum value is 4, so **at the moment** the longest increasing subsequence is **[1, 2, 3, 4]**, **maxLength = 4**, reset currentLength, **currentLength = 1**

. . . Repeat algorithm till the end of an array

The last increasing subsequence will be [1, 2, 3], **currentLength = 3**, but we will not write this value to maxLength variable, brcause we met longer subsequence before. maxLength (4) more than currentLength (3)

So we need to return maxLength and the answer will be **4**