Longest Continuous Increasing Subsequence

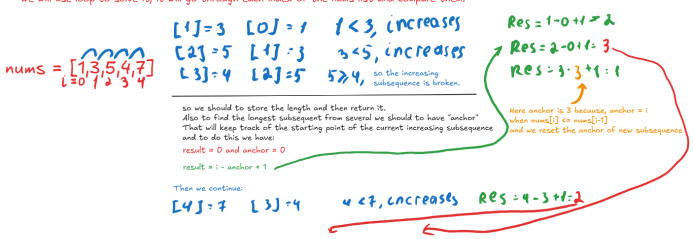
We should firstly understand the problem. So we need to find the length of the longest subsequence.
Subsequense means the part of the array:

nums = [1,3,5,4,7] in this array sucsequense can be [1,3]; [3,5]; [1,3,5]; and so on

So we want to find the longest subsequence in which elements increasing continuous, and we can not move or sort given array.

nums = [1,3,5,4,7] in this case answer will be 3 and subsequense looks like this: nums = [1,3,5]

How we solve this:
We have:
input array: nums = [1, 3, 5, 4, 7]
We will use loop to solve it, it will go through each index of the nums list and compare them.



We get this 2 result and return max

In the end result will be 3

Merge Sorted Array

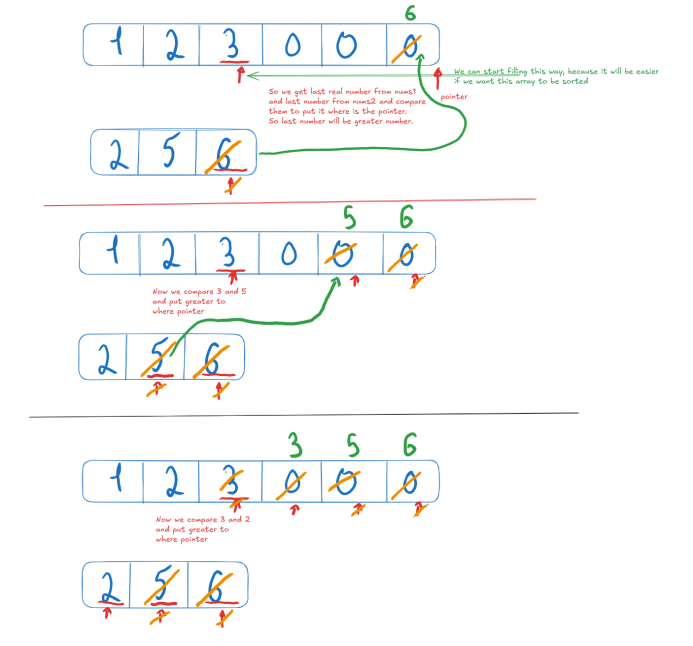
Given two sorted integer arrays nums1 and nums2, we should to merge nums2 into nums1 as one sorted array

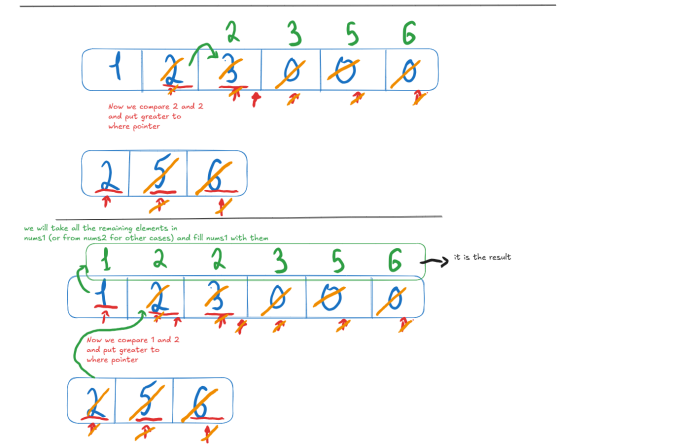
merge

nums1 has a length of m + n, it means that there is enough space in nums1 for elements in nums2 to be inserted

nums1 = [1,2,3,0,0,0]

nums2 = [2,5,6]





Intersection of Two Arrays

Given two integer arrays nums1 and nums2, We should to return an array of their intersection and element in result array must be unique and any order.