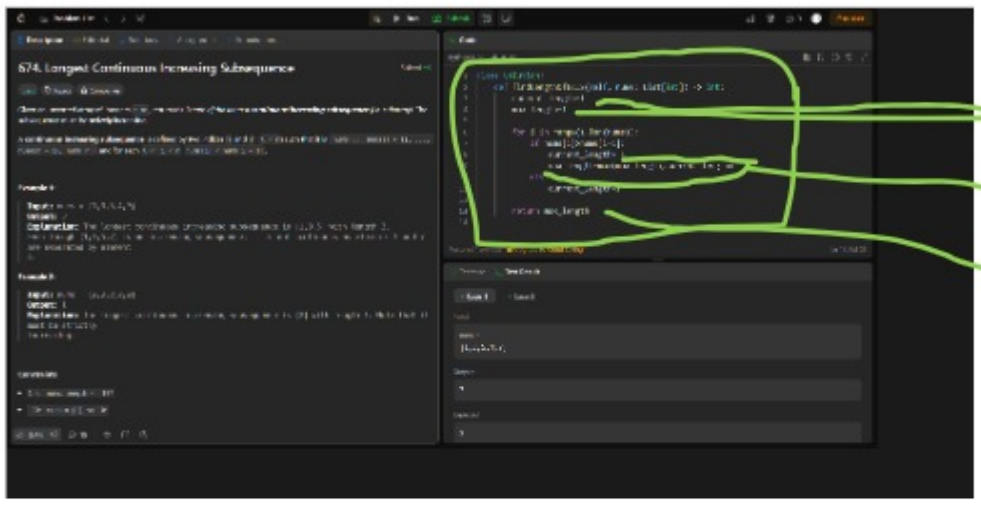
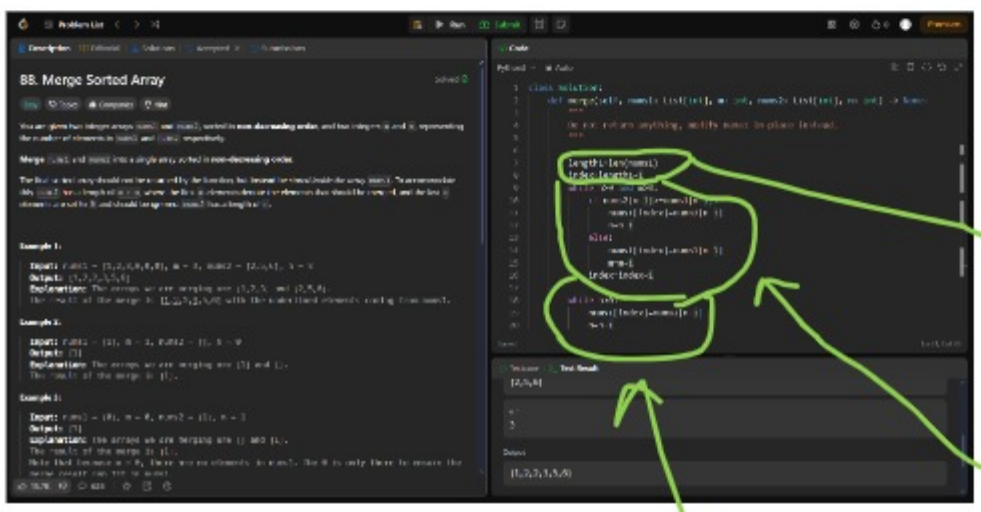


It is my solution to 674. Longest Continuous Increasing Subsequence

Firstly I created two vars for current length of subsequence and for max length subsequence

After i run loop to see and count length of subsequence and compare curr and max to see which one is longer and set it max length subsequence

After code runned all ints in array it will return max length of subsequence

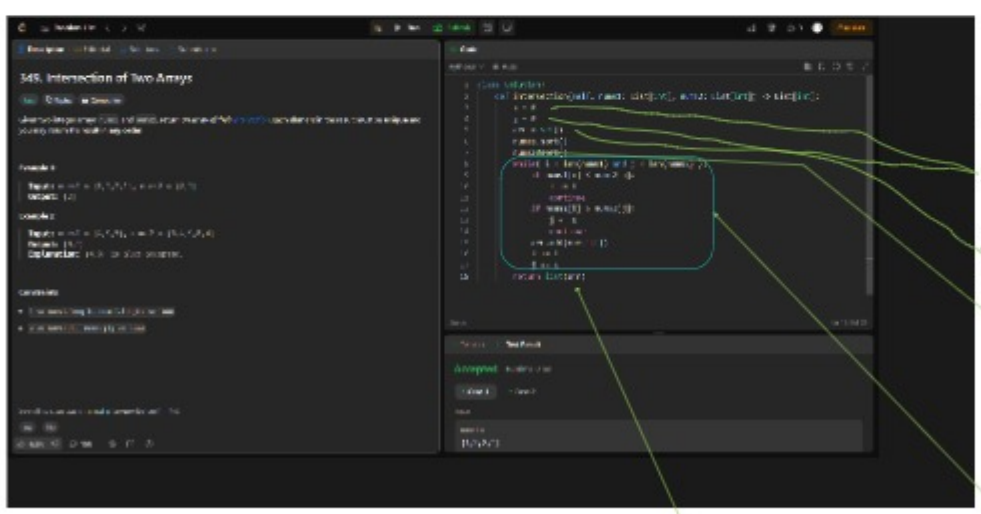


This is solution for 88. Merge Sorted Array

first of all we need index to define where numbers will be placed so index is minus 1 from length of our nums1 because its length is number of elements all two list

next we have two pointers m and n number of element in each list. By looping until one of them not equal to 0 and comparing elements in these lists by these poiters we will fill nums1 and decrease poiters m or n and index

The last part is for cases where m equals 0 and all elements of nums2 will be placed inside of nums1



Here is solution for 349. Intersection of Two Arrays

So here i also used two pointers to check each element in lists

And i created set data structure cause it stores only unique elements

To make it easy i sorted both lists

After i made loop with ywo pointers to run lists and if element of 1 is greater or lower than element of 2 it will skip them with increasing pointers but if neither statments is true element will be added to arr and also increase pointers

At the end i changed set into list to correct result