資料結構

Data Structure

Lab 12

姓名：曾致嘉

學號：113AB0014

Lab12-Q1

You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively.
Merge nums1 and nums2 into a single array sorted in non-decreasing order.
The final sorted array should not be returned by the function, but instead be *stored inside the array* nums1. To accommodate this, nums1 has a length of m + n, where the first m elements denote the elements that should be merged, and the last n elements are set to 0 and should be ignored. nums2 has a length of n.

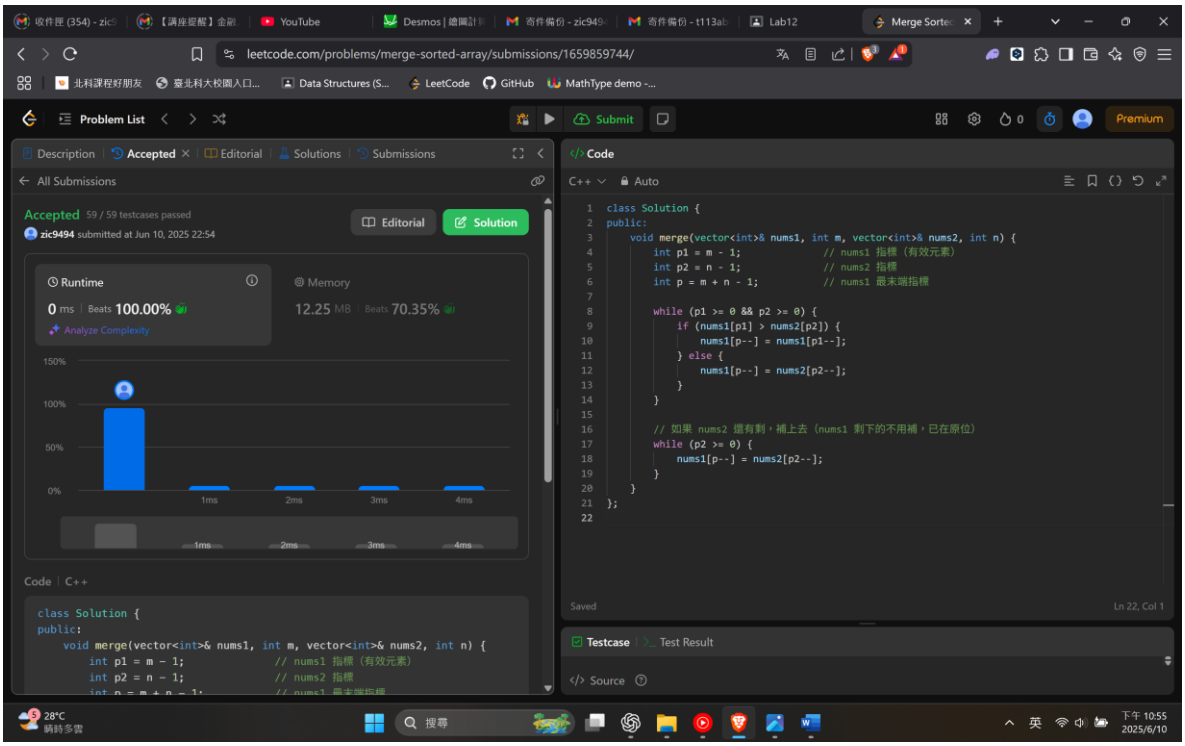
```
#include<iostream>
```

```
class Solution {
public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
        int p1 = m - 1;                    // nums1 指標（有效元素）
        int p2 = n - 1;                    // nums2 指標
        int p = m + n - 1;                 // nums1 最末端指標

        while (p1 >= 0 && p2 >= 0) {
            if (nums1[p1] > nums2[p2]) {
                nums1[p--] = nums1[p1--];
            } else {
                nums1[p--] = nums2[p2--];
            }
        }

        // 如果 nums2 還有剩，補上去（nums1 剩下的不用補，已在原位）
        while (p2 >= 0) {
            nums1[p--] = nums2[p2--];
        }
    }
};
```

Discussion Section



複雜度：$O(m+n)$

邏輯：

將新的資料比較中比較大的一直放到陣列的最後方