Problem 01: Anagram Sorting

• Time complexity: Suppose there are n strings in the input array, and the average length of each string is m. For sorting algorithms, a common time complexity is O(n \* m \* log m), because the time complexity of sorting depends on the average length of the string. The space complexity is O(n \* m) because the sorted string needs to be stored.

Problem 02: Sorted Array Merging

• Time complexity: Assume that the size of array A is m, the size of array B is n, and m is greater than or equal to n. The time complexity of merging two ordered arrays is O(m + n), because it is necessary to traverse the two arrays All elements at once. The space complexity is O(1), because the operation is performed on the original array A, and no additional space is required.

Problem 03: Pascal's Triangle

• Time complexity: Suppose we calculate the first n rows of Pascal's Triangle, where n is larger. The time complexity of calculating each row is about O(n2), because each row has n elements, and the calculation of each element requires O(N) time. The total time complexity is about O(n3). The space complexity is O(1) because only a small amount of space for variables and constants needs to be stored.