689. Maximum Sum of 3 Non-Overlapping Subarrays
We will use dp to solve this question. We will get first parts max index and third part's max index. Then we will compare mid + max_first + max_third to get the maximum combination. TC is O(n), SC is O(n)

```
class Solution:
    def maxSumOfThreeSubarrays(self, nums: List[int], k: int) -> List[int]:
        left, right, length = [], [], len(nums)
        sums = [0]
        total = 0
        ans = [0,0,0]
        for i in nums:
            sums.append(sums[-1] + i)
        for i in range(0, length - 3 * k + 1):
            if total < sums[i + k] - sums[i]:
                left.append(i)
                total = sums[i + k] - sums[i]
            else:
                left.append(left[-1])
        total = 0
        for i in range(length - k, 2 * k - 1, -1):
            if total <= sums[i + k] - sums[i]:
                right.append(i)
                total = sums[i + k] - sums[i]
            else:
                right.append(right[-1])
        right.reverse()
        total = 0
        for i in range(k, length - 2 * k + 1):
            l = left[i - k]
            r = right[i - k]
            if total < sums[l + k] - sums[l] + sums[r + k] - sums[r] + sums[i + k] - sums[i]:
                ans = [l, i, r]
                total = sums[l + k] - sums[l] + sums[r + k] - sums[r] + sums[i + k] - sums[i]
        return ans
```

273. Integer to English Words

```
class Solution:
    def numberToWords(self, num: int) -> str:
        LESS_THAN_20 = ["", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"]
        TENS = ["", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"]
```

```
        THOUSANDS = ["", "Thousand", "Million", "Billion"]

        def helper(num):
            if num == 0:
                return ""
            elif num < 20:
                return LESS_THAN_20[num] + " "
            elif num < 100:
                return TENS[num // 10] + " " + helper(num % 10)
            else:
                return LESS_THAN_20[num // 100] + " Hundred " + helper(num % 100)

        if num == 0:
            return 'Zero'
        i, words = 0, ""
        while num > 0:
            if num % 1000 != 0:
                words = helper(num % 1000) + THOUSANDS[i] + " " + words
            num = num // 1000
            i += 1
        return words.strip()
```

349. Intersection of Two Arrays

```
class Solution:
    def intersection(self, nums1: List[int], nums2: List[int]) -> List[int]:
        return list(set(nums1) & set(nums2))
```

350. Intersection of Two Arrays II

```
from collections import Counter
class Solution:
    def intersect(self, nums1: List[int], nums2: List[int]) -> List[int]:
        counter1 = Counter(nums1)
        counter2 = Counter(nums2)
        result = []
        for k, v in counter1.items():
            if k in counter2:
                result.extend([k] * min(v, counter2[k]))
        return result
```

88. Merge Sorted Array

We will start from tail and TC is O(n)

```
class Solution:
    def merge(self, nums1: List[int], m: int, nums2: List[int], n: int) -> None:
        """
        Do not return anything, modify nums1 in-place instead.
        """
        length = m + n
        idx1, idx2 = m - 1, n - 1
        idx = m + n - 1
        while idx >= 0 and idx1 >= 0 and idx2 >= 0:
            if nums1[idx1] > nums2[idx2]:
                nums1[idx] = nums1[idx1]
                idx1 -= 1
            else:
                nums1[idx] = nums2[idx2]
                idx2 -= 1
            idx -= 1

        while idx >= 0 and idx1 >= 0:
            nums1[idx] = nums1[idx1]
            idx1 -= 1
            idx -= 1

        while idx >= 0 and idx2 >= 0:
            nums1[idx] = nums2[idx2]
            idx2 -= 1
            idx -= 1
```