Assignment #5

1. **Burst Balloons**

Given n balloons, indexed from 0 to n-1. Each balloon is painted with a number on it represented by array nums. You are asked to burst all the balloons. If the you burst balloon i you will get nums[left] * nums[i] * nums[right] coins.
Here left and right are adjacent indices of i. After the burst, the left and right then becomes adjacent.

Find the maximum coins you can collect by bursting the balloons wisely.

Note:
(1) You may imagine nums[-1] = nums[n] = 1. They are not real therefore you can not burst them.
(2) 0 ≤ n ≤ 500, 0 ≤ nums[i] ≤ 100

Example:
Given [3, 1, 5, 8]
Return 167
nums = [3,1,5,8] --> [3,5,8] --> [3,8] --> [8] --> []
coins = 3*1*5 + 3*5*8 + 1*3*8 + 1*8*1 = 167

2. Count of Smaller Numbers After Self

You are given an integer array nums and you have to return a new counts array. The counts array has the property where counts[i] is the number of smaller elements to the right of nums[i].

Example:
Given nums = [5, 2, 6, 1]

To the right of 5 there are 2 smaller elements (2 and 1).
To the right of 2 there is only 1 smaller element (1).
To the right of 6 there is 1 smaller element (1).
To the right of 1 there is 0 smaller element.

Return the array [2, 1, 1, 0].

3. Count of Range Sum

Given an integer array nums, return the number of range sums that lie in [lower, upper] inclusive.

Range sum S(i, j) is defined as the sum of the elements in nums between indices i and j (i ≤ j), inclusive.

Note:
A naive algorithm of O(n2) is trivial. You MUST do better than that.

Example:
Given nums = [-2, 5, -1], lower = -2, upper = 2,
Return 3.
The three ranges are : [0, 0], [2, 2], [0, 2] and their respective sums are -2, -1, 2