2023-09-23 - Handout – Sorting

Q1. Valid Triangle Number

Link: https://leetcode.com/problems/valid-triangle-number/

Given an integer array nums, return the number of triplets chosen from the array that can make triangles if we take them as side of lengths of a triangle.

Example1
Input: nums = [2, 2, 3, 4]
Output: 3
Explanation: Valid combinations are:
2,3,4 (using the first 2),
2,3,4 (using the second 2)
2,2,3

Example2:
Input: nums = [4,2,3,4]
Output: 4

Constraints:
1 <= nums.length <= 1000

Q2. Shortest Unsorted Continuous Subarray

Link: https://leetcode.com/problems/shortest-unsorted-continuous-subarray/description/

Given an integer array nums, you need to find one continuous subarray such that if you only sort this subarray in the non-decreasing order, then the whole array will be sorted in non-decreasing order. Return the shortest such subarray and output its length.

Example1
Input: nums = [2 ,6, 4, 8, 10, 9, 15]
Output: 5
Explanation: You need to sort [6, 4, 8, 10, 9] in ascending order to make the whole array sorted in ascending order.

Example2
Input: nums = [2, 3, 4, 1, 10, 7, 8]
Output: 7

Constraints:
1 <= nums.length <= 10^4
-10^5 <= nums[i] <= 10^5

Q3. Count of Smaller numbers After Self

Link: https://leetcode.com/problems/count-of-smaller-numbers-after-self/description/

Given an integer array nums, return an integer array *counts* where *counts[i]* is the number of smaller elements to the right of nums[i].

Example1
Input: nums = [5,2,6,1]
Output: [2,1,1,0]
Explanation:
To the right of 5, there are 2 smaller elements (2 and 1)
To the right of 2, there is 1 smaller element (1)
To the right of 6, there is 1 smaller element (1)
To the right of 1, there is no smaller elements.

Example2
Input: nums = [-1]
Output: 0

Constraints:
$1 <= nums.length <= 10^5$
$-10^4 <= nums[i] <= 10^4$

Q4. Count of Range Sum

Link: https://leetcode.com/problems/count-of-range-sum/

Given an integer array **nums** and two integers *lower* and *upper*, 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 inclusive, where i <= j.

Example1
Input: nums = [-2, 5, -1], lower = -2, upper = 2
Output: 3
Explanation: The three ranges are: [0, 0], [2, 2], and [0, 2] and their respective sums are: -2, -1, 2

Example2
Input: nums = [0], lower = 0, upper = 0
Output: 1

Constraints:
$1 <= nums.length <= 10^5$
$-2^{31} <= nums[i] <= -2^{31} -1$
$-10^5 <= lower <= upper <= 10^5$
The answer is guaranteed to fit in a 32-bit integer.