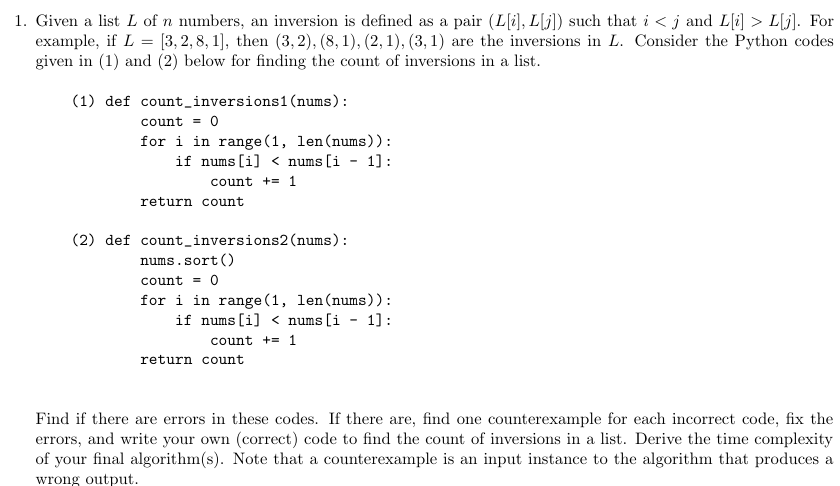
**ASSIGNMENT – 3**

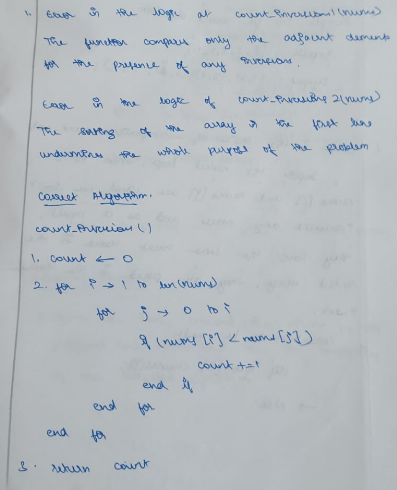
**AIM:**

To find the errors in the given code or algorithm rectify the error using the best approach

**Qn1:**

****

**Psuedo Code:**

****

**Source Code:**

def count\_inversions(nums):

    count = 0

    inversion = []

    for i in range(1, len(nums)):

        j = i-1

        while j>=0:

            if nums[i] < nums[j]:

                inversion.append((nums[j], nums[i]))

                count += 1

            j -= 1

    return count, inversion

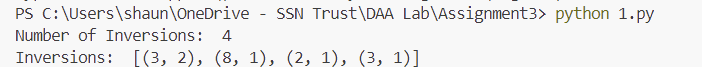
l = [3,2,8,1]

count, inversion = count\_inversions(l)

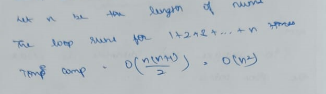
print("Number of Inversions: ", count)

print("Inversions: ", inversion)

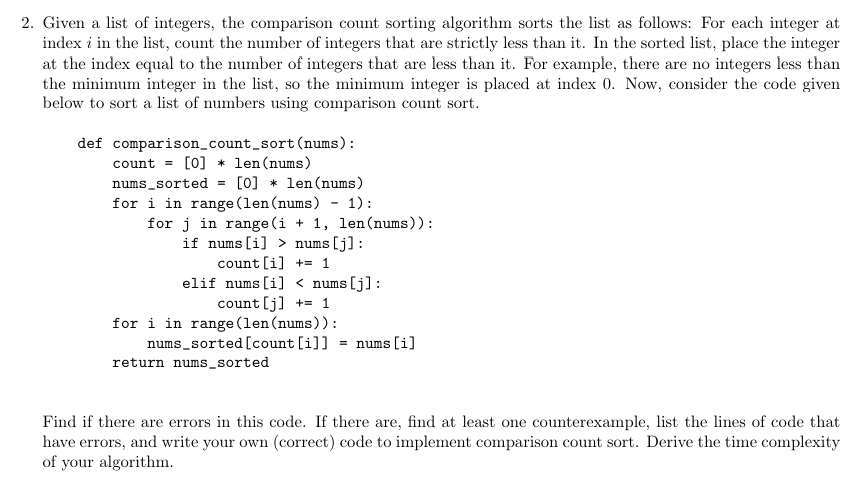
**Output:**

****

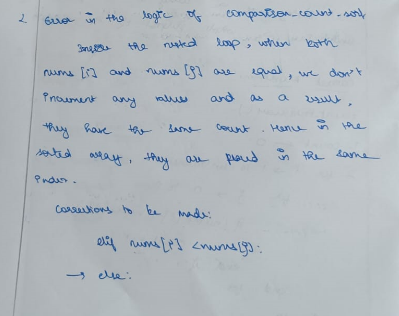
**Time Complexity:**

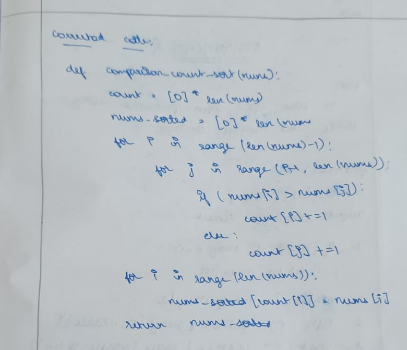
****

**Qn2:**

****

**Psuedo Code:**

****

****

**Source Code:**

def comparison\_count\_sort(nums):

    count = [0] \* (max(nums)+1)

    nums\_sorted = [0] \* len(nums)

    for i in range(len(nums) - 1):

        for j in range(i + 1, len(nums)):

            if nums[i] > nums[j]:

                count[nums[i]] += 1

            elif nums[i] < nums[j]:

                count[nums[j]] += 1

    for i in nums:

        nums\_sorted[count[i]] = i

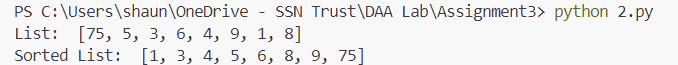
    return nums\_sorted

l = [75,5,3,6,4,9,1,8]

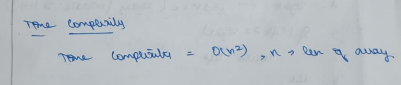
print("List: ", l)

print("Sorted List: ", comparison\_count\_sort(l))

**Output:**

****

**Time Complexity:**

****

**Learning Outcomes:**

* I learnt to analyse the errors in codes of various algorithms
* I learnt how to implement various sorting and searching algorithms in Python