CS 560 HW 2: Asymptotic Notations and Maximum-Subarray Problem

Note: this homework assignment has three questions.

(Apply to all Huabo's CS 560 homework assignments) Please try to solve them yourself without referring to solutions posted online or ChatGPT.

In the case you copy a solution from online resources or ChatGPT, please comply with the following reflection requirements:

- Please acknowledge so.
- State any other attempts you tried to solve before referencing the online solution.
- Summarize your comprehension after referencing the online solution.

You will not be penalized for referring to online solutions AND complying with the required reflection. Failure to comply with the required reflection will result in a homework grade penalty.

Question 1

Solve (again) CLRS Exercise 3.1-1, the max(f(n), g(n)) = Θ(f(n) + g(n)) question.

Question 2

Use Big Oh's definition to prove that f(n) = n*n + 10n +101 = O(n*n).

Question 3

Improve the following function so the time complexity will be reduced from O(n*n*n) to O(n*n)

```
def naiveMaxSubarray(arr:list)->list:
    # two nested loops to compare all possible pairs

    # assume the arr is not empty. Otherwise, there is no max subarray
    currentMax = arr[0]
    i_max, j_max = 0, 0
    n = len(arr)
    for i in range(n):
        # TODO: get rid of the sum function for better time complexity
        for j in range(i,n):
            if sum(arr[i:j]) > currentMax:
                currentMax = sum(arr[i:j])
                i_max, j_max = i, j
    return [i_max, j_max-1]
```