DSC 40B - Discussion 02

Problem 1.

Compute the best and worst case time complexity for the following code snippets:

a)
```
def f_1(arr1, arr2):
    """`arr1` and `arr2` are two arrays each of size n"""
    n = len(arr1)
    for i in range(n):
        for j in range(n):
            if arr[i] + arr[j] == 0:
                return (i,j)
```

b)
```
def insertion_sort(arr):
    """Sort `arr` in ascending order."""
    n = len(arr)
    for i in range(1, n):
        x = arr[i]
        j = i - 1
        # find where to place x
        while j >= 0 and x < arr[j]:
            arr[j+1] = arr[j]
            j -= 1
        arr[j+1] = x
```

Problem 2.

Compute the average time complexity for the following code snippet:

a)
```
def f_1(arr1, arr2):
    """`arr1` and `arr2` are two arrays each of size n"""
    n = len(arr1)
    for i in range(n):
        for j in range(n):
            if arr[i] + arr[j] == 0:
                return (i,j)
```

Problem 3.

Provide a tight theoretical lower bound for the problems given below. Provide justification for your answer.

a) Given an array of n numbers, find the sum of the numbers in the array.

b) Given a sorted array of $n \geq 2$ numbers, find the second largest number in the array.

Problem 4.

Provide the asymptotic time complexity of the operation below using multivariate Θ notation.
Given two d-dimensional vectors $v = [v_1, v_2, ..., v_d]$ and $u = [u_1, u_2, ..., u_d]$, their sum is computed as $v + u = [v_1 + u_1, v_2 + u_2, ..., v_d + u_d]$. Given n vectors over R^d, check if any two of them sum to the zero vector using the brute force method.