Assignment 1

1. Order the following list of functions by the big-O notation from highest to lowest. Total functions are twelve to arrange. It's a single problem. Don't take it as two problems.

$n \log n$, $\log \log n$, $1/n$, $4n^{3/2}$, $5n$, $2n \log 2n$, 2^n, 4^n, n^3, $n^2 \log n$, $4^{\log n}$, $n^{1/2}$

Answer: Factorial➔Exponential➔Cubic➔Quaadratic➔N x log N➔Linear➔Log➔constant

4^n, 2^n, $4^{\log n}$, n^3, $4n^{3/2}$, $n^2 \log n$, $2n \log 2n$, $n \log n$, $5n$, $\log \log n$, $n^{1/2}$, $1/n$,

Rojin ->

$1/n$, $\log \log n$, $n \log n$, $5n$, $4n^{3/2}$, $n^{1/2}$, $4^{\log n}$, $2n \log 2n$, , $n^2 \log n$, n^3 , 2^n , 4^n

2. Give a big-O characterization, in terms of *n*, of the running time of the Loop1 method below:

```
Algorithm Loop1(n)
        s ← 0
          for i ← 1 to n do
            s ← s + i
```

Answer: O(n), because i will go up to n.

3. Perform a similar analysis for method Loop2 below:

```
Algorithm Loop2(n)
        s ← 0
          for i ← 1 to n^2 do
             for j ← 1 to i do
                  s ← s + i
```

Answer:

Outer loop goes up to n^2 and inner loop will go up to n^2 since j goes till i. So overall worst-case complexity would be O(n^4).

4. Decide whether each of the following is true or false. Justify your answer why it is true or false.

a. $\log n$ is $O(n)$ [Take $F(n) = \log n$, $g(n) = n$]
b. 2^n is $O(n^2)$ [Take $f(n) = 2^n$, $g(n) = n^2$]

Answer:

The statement a is true because f(n) is O(g(n)) where g(n) is an asymptotic upper bound on f(n). Also, g(n) grows faster than f(n).

The statement b is false because f(n) is an exponential function which grows more than quadratic function g(n).

5. The given code will find the sum of n elements for the array. Your job is to analyse the best and worst case runtime for this given function.

```
function sumArray(arr) {
  if (arr.length === 1) {
    return arr[0];
  }
```

```
    let total = 0;
    for (let i = 0; i < arr.length; i++) {
        total += arr[i];
    }
    return total;
}
```

Answer:

Best Case(Scenario) & O(?)
O(1)

Worst Case(Scenario) & O(?)
O(n)

6. Write a Pseudo code to return the count of even numbers in the given array and analyse the Big O runtime for this algorithm.

Interview Question is not necessary to submit, but recommend to present in the afternoon Lab sessions.

Interview Question, you need to solve this given problem and present solution in the lab session.

Problem: Given an array of integers nums and an integer target, return *indices of the two numbers such that they add up to target*.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2:

Input: nums = [3,2,4], target = 6
Output: [1,2]

Example 3:

Input: nums = [3,3], target = 6

Output: [0,1]

[**Note:** if you return [0,0] is wrong answer, may not use the same element twice]

What is the time complexity of your solution.

O(n^2)

```
function addUp(arr: number[], totalSum: number): number[] {
let currentNum: number, otherNum: number;
for (let i = 0; i < arr.length; i++) {
  currentNum = arr[i];
  for (let j = i + 1; j < arr.length; j++) {
```

```
        otherNum = arr[j];
        if (currentNum + otherNum === totalSum) {
          return [i, j];
         }
    }
}
return [];
}
```