

**CS2413: Data Structures
Fall 2021**

Homework #1

- Full nameonly: Bibek Dhungana
- Release date: Aug 27th, 2021 (Friday)
- Due date: **Sept 2nd, 2021 (Thursday) before midnight, 11:59 PM**
- It should be done INDIVIDUALLY; Show ALL your work; Write your answer in a Word file and submit it through the blackboard
- Total: 10 pts

1. Explain the meaning of the expression, $f(n)$ is $O(1)$.

[3 pts]

A function represented by Big O notation provides information about upper bound/growth rate of the function.

In this case, $f(n)$ is $O(1)$ means the constant. It means, the algorithm takes fixed number of steps regardless of size of problems. It will take constant time/space to solve the problem independent of n .

For e.g., In array, elements are stored in continuous memory location. No matter how long the array is, the first element of the array can be accessed in constant time. so, algorithmic complexity of accessing first element of array is constant.

2. Analyze the running time and find the computational complexity (Big-Oh will do) of the following code. Show all your work.

[4 pts]

```
for (cnt2 = 0, i = 1; i <= n; i ++)  
    for (j = 1; j <= i; j ++)  
        cnt2 ++;
```

Let us see this code steps by steps.
`cnt2 = 0` is assignment and require constant time.
`i = 1` is assignment and require constant time.

Outer loop is started from $i = 1$ and repeated until $i \leq n$. so, the outer loop is repeated n times. (linear)

Inside loop.
The inside loop is started from 1 and repeated until $j \leq i$. so, it is executed i times.

We need to calculate how many times is `cnt2++` executed.
When $i = 1$, it is executed 1 time.
When $i = 2$, it is executed 2 times.
and so on.
The sum of all-natural number is:
 $1 + 2 + 3 + 4 + \dots + n = n(n+1)/2$.

Overall assignment becomes:
Hence, overall computational complexity (Big O) is $O(n^2)$.

[3 pts]

3. Order the following functions by growth rate.

2^n 1500 $n \lg n$ n^3 $\lg n$

The order of the growth rate of the following functions are:

1. 1500 - constant
2. $\lg n$ - logarithmic
3. $n \lg n$ - log-linear
4. n^3 - polynomial (cubic)
5. 2^n - exponential

Hence, $1500 < \lg n < n \lg n < n^3 < 2^n$ is increasing order of growth rate.