

# Lab 3

## Problem 1:

$$1. \quad 10000000000n^2 \quad O(n^2) < O(n^3) \quad \checkmark$$

$$2. \quad \begin{array}{l} n^2 \log n \quad \checkmark \\ \log(n^2 \log n) \\ \log n^2 + \log \log n \\ 2 \log n + \log \log n \end{array} \quad \begin{array}{l} n(\log n^{10}) \\ \log(n \log n^{10}) \\ \log n + \log \log n^{10} \\ \log n + 10 \log \log n \end{array}$$

$$3. \quad \begin{array}{l} n^{\log n} \quad \checkmark \\ \log n^{\log n} \\ \log n \log n \\ \log^2 n \\ 2 \log n \end{array} > \begin{array}{l} 2^{\sqrt{n}} \\ \log 2^{\sqrt{n}} \\ \sqrt{n} \log 2 \\ \sqrt{n} \\ \frac{1}{2} \log n \end{array}$$

$$4. \quad \begin{array}{l} 2^n \\ \log 2^n \\ n \log 2 \\ n \end{array} < \begin{array}{l} 2^{2n} \quad \checkmark \\ \log 2^{2n} \\ 2n \log 2 \\ 2n \end{array}$$

However, in terms of asymptotic notation, both would be  $O(n^2)$ .

## Problem 2:

1. The best case would be when the input  $n$  would be <sup>less than 4</sup> 1 in which case it would take  $\Theta(1)$  time.
2. The worst case would be when the input is a prime number, in which case it would take  $\Theta(\sqrt{n})$  time.
3. The average case is not very straightforward to calculate, but in the majority of cases, the function will need to check until the square root of a number to ascertain if it is prime. Hence, the average case time complexity will stay at  $\Theta(\sqrt{n})$ .

The time complexity would be  $\Theta(\sqrt{n})$ .