S2021 COMS311 Recitation Week 2
Sections 1 4

TA: Alvin Chon

Please refer to, Recitation Notes by Weisi, for the first portion of the recitation notes. I will almost always try to use the common recitation materials first and then may augment them with more material.
This recitation is about Big-O and time complexity. Make sure you know it and can use it both for analyzing an algorithm given to you and in making your own.

1 Problem1

1.1 a

$f(n) = n$. Is $f(n) = O(log(n))$? Why or why not?

1.2 c

$f(n) = n$. Is $f(n) = O(n^2)$? Why or why not?

1.3 b

$f(n) = n$. Is $f(n) = O(n^{999})$? Why or why not?

1.4 d

Prove that $x^2 + 10x + 1000$ is $O(x^2)$.

1.5 e

Disprove that x^3 is $O(x^2)$.

2 Problem2

Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, ..., g_{30}$ of the functions satisfying $g_1 = O(g_2), g_2 = O(g_3), ..., g_{29} = O(g_{30})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$.
Scroll down to the next page for the answer.

$2^{lgn},\ 2^{2^n},\ e^n,\ n^2,\ 1,\ 4^{lgn},\ 9n,\ 2^n,\ 2^{2^n+1},\ n!,$

$$1 < 9n = 2^{lgn} < n^2 = 4^{lgn} < 2^n < e^n < n! < 2^{2^n} < 2^{2^n+1}$$

Remember that $lg(n) \neq log(n)$ and $lg(n) = log_2(n)$ and thus $2^{lg(n)} = n$