CSE 450
Assignment 1

31^{th} August, 2022

Submission Instructions: Deadline is **11:59pm on 9/7/2022**. Late submissions will be penalized, therefore please ensure that you submit (file upload is completed) before the deadline. Additionally, you can download the submitted file to verify if the file was uploaded correctly. **Please TYPE UP YOUR SOLUTIONS and submit a PDF** electronically, via *Canvas.*
Furthermore, please note that the graders will grade 2 out of the 4 questions randomly. Therefore, if the grader decides to check questions 1 and 4,and you haven't answered question 4, you'll lose points for question 4. Hence, please answer all the questions.

1. Take the following list of functions and arrange them in ascending order of growth rate. That is, if function g(n) immediately follows f(n) in your list, then it should be the case that f(n) is O(g(n)). **(25)**
   (a) $f_1(n) = 2^{100}$
   (b) $f_2(n) = n^2$
   (c) $f_3(n) = nlog(n)$
   (d) $f_4(n) = n$
   (e) $f_5(n) = 2^n$
   (f) $f_6(n) = log(n^2)$
   (g) $f_7(n) = 2^{log(n)}$
   (h) $f_8(n) = n^n$

   Solution: $f_1 < f_6 < f_7 < f_4 < f_3 < f_2 < f_5 < f_8$

2. Prove or disprove the following with valid arguments: **(5+5+5+5+5)**
   (a) $3n^2 + 5 \in O(n)$
   (b) $n^{1.1} \in O(n)$
   (c) $n^2 log(n) \in \Theta(n^2)$
   (d) $2^n \in O(n!)$
   (e) $O(n^4) + \Theta(n^3) \in \Theta(n^4)$

Solution:
(a) FALSE.
Suppose that $3n^2 + 5 \in O(n)$, which means that we can find $c > 0$ and $n_0 >= 1$ such that
$3n^2 + 5 <= c \times n$,
Dividing both sides by n, we got $3n + 5/n <= c$
Since the LHS is a function of n, we cannot find a constant which will satisfy the above inequality.
Therefore, FALSE that $3n^2 + 5 \in O(n)$

(b) FALSE.
Suppose that $n^{1.1} \in O(n)$, which means that we can find $c > 0$ and $n_0 >= 1$ such that
$n^{1.1} <= c \times n$,
Dividing both sides by n, we got $n^{0.1} <= c$
The above inenquality can not be true since c must be a constant but $n^{0.1}$ is unbounded. In fact, as soon as n $> e^{10log(c)}$ we have c $< n^{0.1}$
This is a contradiction with the assumption that we can find such a constant c.
Therefore, FALSE that $n^{1.1} \in O(n)$

(c) FALSE.
Case 1: $n^2 log(n) \in \Omega(n^2)$
$n^2 log(n) >= c_1 \times n^2$
$log(n) >= c_1$, after dividing both sides by n^2,
Selecting $c_1 = 1$ for $n > 1$ will satisfy the above equation. Therefore $n^2 log(n) \in \Omega(n^2)$

Case 2 : $n^2 log(n) \in O(n^2)$
$n^2 log(n) <= c_2 \times n^2$
$log(n) <= c_2$, after dividing both sides by n^2,
Since the LHS is a function of n, we cannot find a constant which will satisfy the above inequality. Therefore, $n^2 log(n)$ is not $O(n^2)$, and by extension, FALSE that (c) $n^2 log(n) \in \Theta(n^2)$

(d) TRUE.
Choose $c = 2$ and $n_0 = 2$. Then for all $n >= n_0$, we have
$2^n = 2 \times 2^{n-1} <= c \times n!$
This proves that $2^n \in$ O$(n!)$

(e) FALSE.
In order to prove this, we only need a counter-example. Let $f(n) = n^3$. Then $f(n)$ is an element of the LHS. However, $f(n)$ is not an element of the RHS, as $\lim_{n\to\infty} n^3/n^4 = 0$.
Therefore, FALSE that $O(n^4) + \Theta(n^3) \in \Theta(n^4)$

3. Prove or disprove the following assertions: **(6+6+6+7)**

   (a) $f(n) = O(f(n)^2)$
   (b) $f(n) = \Theta(max(f(n), g(n))$
   (c) If $f(n) = \Omega(g(n))$ then $f(n) = O(g(n))$
   (d) If $f(n) = O(g(n)$ then $2^{f(n)} = O(2^{g(n)})$

   Solution:
   (a) FALSE. Counter Example : $f(n) = 1/n$
   (b) FALSE. Counter Example : $f(n) = n$, $g(n) = n^2$
   (c) FALSE. Counter Example : $f(n) = n^2$, $g(n) = n$
   (d) FALSE. Counter Example : $f(n) = 2n$, $g(n) = n$

4. Suppose that you have algorithms with the size running times listed below.Assume that these are the exact number of operations performed as a function of the input size n. Suppose you have a computer that can perform 10^{10} operations per second, and you need to compare a result in at most an hour of computation. For each of the algorithms, what is the largest input size n for which you would be able to get the result within an hour? **(5+5+5+5+5)**

   (a) n^2
   (b) n^3
   (c) $100n^2$
   (d) $nlog(n)$
   (e) 2^n

   Solution: Number of operations done in 1 hour = 60 * 60 * 10^{10}.

   (a) n^2
   Solving for $n^2 = 60 * 60 * 10^{10}$, we get,
   $n = 6 \times 10^6$.

   (b) n^3
   Solving for $n^3 = 60 * 60 * 10^{10}$, we get,
   $n = 33019$.

   (c) $100n^2$
   Solving for $100n^2 = 60 * 60 * 10^{10}$, we get,
   $n = 6 \times 10^5$.

   (d) $nlog(n)$
   Solving for $nlog(n) = 60 * 60 * 10^{10}$, we get,

$n = 288,906 \times 10^6$. The result can be differ by various calculation tool

(e) 2^n
Solving for $2^n = 60 * 60 * 10^{10}$, we get,
$n = 45$