Instructor Name: Siddhartha Nath
Email: s.nath@northeastern.edu

Northeastern University

Program Structure and Algorithms (INFO 6205)
Homework #1 – SAMPLE SOLUTIONS – 100 points

Student NAME:

Student ID:

Notes:

- Please submit two files.
- The first file MUST be a PDF that contains your solutions to all questions except the coding question.
- The second file is your solution to the coding question with either .py or .cpp or .java extension.

Question 1 (25 points). *Please prove the following with regards to asymptotic growth of functions.*

(a) *(5 points) Show that* $f(x) = x^2 + 4x$ *is* $O(x^2)$.

Let $g(x) = x^2$, *we have* $x^2 + 4x \leq 1x^2 + 4x^2 \quad \forall x \geq 1$. *Thus,* $c = 5$ *and* $n_0 = 1$.

(b) *(5 points) Show that* $f(x) = x^2$ *is NOT* $O(\sqrt{x})$.

Let $g(x) = \sqrt{x}$, *we have* $x^2 \leq c \cdot \sqrt{x}$. *We cannot find any* $c > 0$ *and* $n_0 \geq 1$ *that can satisfy this inequality as* x^2 *will always grow faster than* $\sqrt{x}$ *for any value of* $x > 1$.

(c) *(5 points) Show that* $f(x) = x$ *is* $\Omega(\log x)$.

Let $g(x) = \log x$, *we have* $x \geq c \cdot \log x$. *If we choose* $c = 1$ *and* $n_0 \geq 1$, *this inequality is satisfied.*

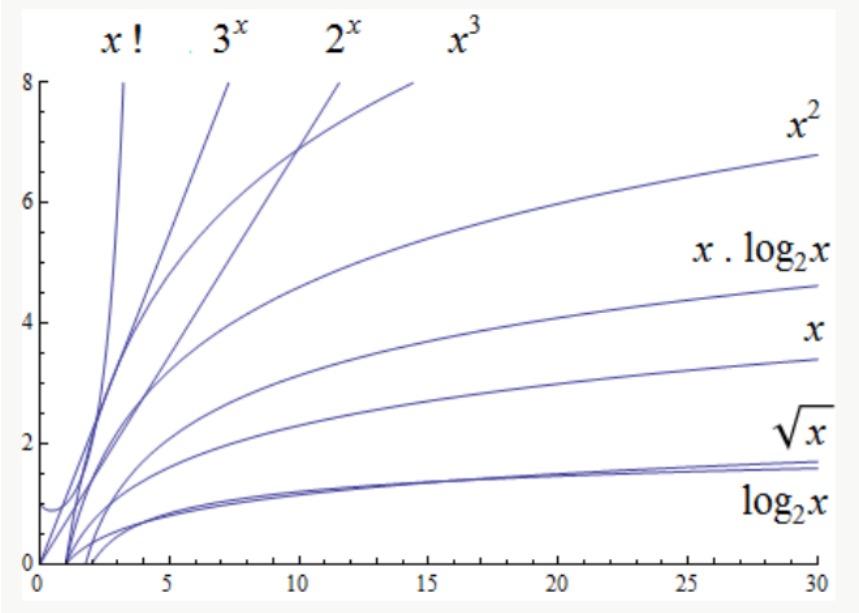
(d) *(10 points) Show that* $f(x) = (2x^2 - 3)/((3x^4 + x^3 - 2x^2 - 1)$ *is* $\Theta(x^{-2})$.

In $f(x)$, *the denominator is* $\Theta(x^4)$, *whereas the numerator is* $\Theta(x^2)$. *Thus,* $f(x) \in \Theta(x^{-2})$.

Question 2 (15 points). *Please rank the following functions based on their $O(\cdot)$ complexity of running time. The function that has the least complexity should be ranked 1. Please explain your answer to get full credit.*

$f_1(x) = x \log_2 x$
$f_2(x) = 3^x$
$f_3(x) = \sqrt{x}$
$f_4(x) = x!$
$f_5(x) = 2^x$

$f_3(x) = Rank\ 1$
$f_1(x) = Rank\ 2$
$f_5(x) = Rank\ 3$
$f_2(x) = Rank\ 4$
$f_4(x) = Rank\ 5$



Question 3 (60 points). *Suppose you are given a string consisting of alphanumeric and parenthesis characters as input. Your goal is to determine if all the open-parenthesis have a corresponding close-parenthesis when you reach the end of the string. If yes, then your algorithm should return True, else False.*

For example, if the input is "I { love [the {rains}()]}", then the output is True. Whereas, if the input is "I { love [the {rains] ()", then the output is False.

(a) *(15 points) Please describe an efficient algorithm in English using a data structure such as array / linked list / stack / queue to solve this problem.*

An English description of the steps are as follows.

1. *Create a stack.*
2. *Iterate over each character in the reversed string.*
3. *If the character is "(" or "{" or "[", push it into the stack.*
4. *If the character is ")" or "}" or "]", pop the stack and check if characters match. If stack is empty or characters don't match then return False.*
5. *Return True if stack is empty.*

(b) *(5 points) What is the asymptotic upper bound of complexity of running time for your algorithm?*

$\Theta(n)$ *if* n *is the size of input string.*

(c) *(40 points) Please write a program in either Python / Java / C++ that realizes your algorithm in (a). To receive full credit, please structure your code, write comments and show the output for the above two examples.*