CS 6363: Design and Analysis of Algorithms – Fall 2019
Homework #1 – Due: Sept. 4
Professor D.T. Huynh

Problem #1. Do Problem # 2-4 (Inversions), p. 41 [Text].

Problem #2. Compare the following pairs of functions $f(n), g(n)$ in terms of order of magnitude. In each case determine whether $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$, and/or $f(n) = \Theta(g(n))$: (Give a brief justification for your answers.)

1. $f(n) = 0.1n^{1.005} + \log\log n$, $g(n) = 100n + (\log n)^5$
2. $f(n) = (\log\log n)^{\log\log n}$, $g(n) = n^{1.5}/\log n$
3. $f(n) = n^6\ 2^{n+4}$, $g(n) = 4^n$

Problem #3.

1. Show that $c^n = o((\log\log\log n)^n)$ for any constant c
2. Show that $\sum_{i=1}^{n} i^k$ is $\Theta(n^{k+1})$.

Problem #4. Do Problem # 7.4-2, p. 184 [Text].

Problem #5 Do Problem # 4.3-8, p. 87 [Text] (Replace n^2 in recurrence by n.)

Problem #6 Do Problem # 4.3-9, p. 88 [Text]

Problem #7. Solve the following recurrences using the method of iteration: (You may use asymtotic notations.)

1. $T(n) = T(n-3) + n$, where $T(0) = 1$
2. $T(n) = 6T(n/4) + n^2$, where $n = 4^k$ and $T(1) = 1$

Problem #8. Do Problem # 4.4-9, p. 93 [Text]

Problem #9. Do Problem # 6.5-9, p.166 [Text]