CSE 4310 – Worksheet2

Algorithm Analysis- Time Complexity

Big Oh, Omega, Theta

1. (10 pts) Use the definition of Big O notation to find the constants c, n_o which show that T(n) is O(f(n)).
   - $T(n) = 3n^3 + 4,\ f(n) = 5n^2$

   - $T(n) = 2^{n+1},\ f(n) = 2^n$

2. (20 pts) Suppose you have algorithms with the five running times listed below. (Assume these are the exact running times.) How much slower do each of these algorithms get when you (a) double the input size, and (b) increase the input size by one.
For example: consider $T(n) = n^2$, when we double the input size , then $T(n)_{new} = (2n)^2 = 4n^2$ .i.e. the running time is increased by a factor of 4 or quadruple , and if we increase by one , then $T(n)_{new} = (n+1)^2 = n^2 + 2\mathrm{n} + 1$ i.e. the running time increased by a factor of $2n + 1$

- $T(n) = n^3$

- $T(n) = 100n^2$

- $T(n) = 2^n$

- $T(n) = log_2 n$

3. (10 pts) For each of the following pairs of functions, either f(n) is in O(g(n)), f(n) is in Ω(g(n)), or f(n) = Θ(g(n)). Determine which relationship is correct and briefly explain why. Hint : use Lopital's rule

- $f(n) = logn^2;\ g(n) = logn + 5$

- $f(n) = log^2 n;\ g(n) = logn$

- $f(n) = 2^n;\ g(n) = 3^n$

4. (20 pts) Arrange in increasing order of asymptotic growth. All logs are in base 2
Hint: *take the log to base 2 of all functions or compare each of them using Lopital's rule*

- $T_1(n) = n^{5/3}$
- $T_2(n) = 2^{\sqrt[2]{\log n}}$
- $T_3(n) = \sqrt[2]{n^n}$
- $T_4(n) = n^{5/3}$
- $T_5(n) = 2^n$