Advanced Algorithms and Data Structures
Asymptotic Notations
Exercises

Prepared by LY Rottana

1. Given the following functions, sort them in increasing order of *Big-Oh* complexity.
   $f_1(n) = n^{0.999999}\log n$
   $f_2(n) = 10000000n$
   $f_3(n) = 1.000001^n$
   $f_4(n) = n^2$

2. For each of the following functions, prove whether f(n) = O(g(n)), f(n) = Ω(g(n)), or f(n) = Θ(g(n)). For example, by specifying some explicit constants n_0 and c > 0 such that the definition of Big-Oh, Big-Omega, or Big-Theta is satisfied.
   a. $f(n) = n \log(n^3)$ $\quad g(n) = n \log n$
   b. $f(n) = 2^{2n}$ $\quad g(n) = 3^n$
   c. $f(n) = \sum_{i=1}^{n} \log i$ $\quad g(n) = n \log n$

3. One of the two software packages, A or B, should be chosen to process very big databases, containing each up to 10^{12} records. Average processing time of the package A is $T_A(n) = 0.1\, n \log_2 n$ microseconds, and the average processing time of the package B is $T_B(n) = 5\, n$ microseconds. Which algorithm has better performance in a *Big-Oh* sense?

4. Let processing time of an algorithm of *Big-Oh* complexity O(f(n)) be directly proportional to f(n). Let three such algorithms A, B, and C have time complexity $O(n^2)$, $O(n^{1.5})$, and $O(n \log n)$, respectively. During a test, each algorithm spends 10 seconds to process 100 data items. Derive the time each algorithm should spend to process 10,000 items.

5. Assume that each of the expressions below gives the processing time T(n) spent by an algorithm for solving a problem of size n. Select the dominant term(s) having the steepest increase in n and specify the lowest *Big-Oh* complexity of each algorithm.

Expression	Dominant term(s)	$O(\ldots)$
$5 + 0.001n^3 + 0.025n$		
$500n + 100n^{1.5} + 50n \log_{10} n$		
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$		
$n^2 \log_2 n + n(\log_2 n)^2$		
$n \log_3 n + n \log_2 n$		
$3 \log_8 n + \log_2 \log_2 \log_2 n$		
$100n + 0.01n^2$		

$0.01n + 100n^2$		
$2n + n^{0.5} + 0.5n^{1.25}$		
$0.01n \log_2 n + n(\log_2 n)^2$		
$100n \log_3 n + n^3 + 100n$		
$0.003 \log_4 n + \log_2 \log_2 n$		