Exercises #2
Asymptotic Analysis

Theoretical Background

Remember the asymptotic notation:

- $\mathbf{f(n) = O(g(n))}$ if there exist positive constants n_0 and c such that $f(n) \leq cg(n)$ for all $n \geq n_0$.
- $\mathbf{f(n) = \Omega(g(n))}$ if there exist positive constants n_0 and c such that $f(n) \geq cg(n)$ for all $n \geq n_0$.
- $\mathbf{f(n) = \Theta(g(n))}$ if there exist positive constants n_0, c_1 and c_2 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.
- $\mathbf{f(n) = o(g(n))}$ if for any positive constant c there exists n_0 such that $f(n) < cg(n)$ for all $n \geq n_0$.
- $\mathbf{f(n)} = \omega(\mathbf{g(n)})$ if for any positive constant c there exists n_0 such that $f(n) > cg(n)$ for all $n \geq n_0$.

Asymptotic Notation

1. Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$. Justify your answer with brief proofs.

2. For each pair of functions $f(n)$ and $g(n)$, indicate whether $f(n)$ is O, o, Ω, ω, or Θ of $g(n)$. Your answer should be in the form of a "yes" or "no" for each cell of the table.

	$f(n)$	$g(n)$	O	o	Ω	ω	Θ
(a)	$2n^3 - 10n^2$	$25n^2 + 37n$					
(b)	56	$\log_2 30$					
(c)	$log_3 n$	$\log_2 n$					
(d)	n^3	3^n					
(e)	$n!$	2^n					
(f)	$n!$	n^n					
(g)	$n \log_2 n + n^2$	n^2					
(h)	$\sqrt{n}$	$\log_2 n$					
(i)	$\log_3(\log_3 n)$	$\log_3 n$					
(j)	$\log_2 n$	$\log_2 n^2$					

3. For each of the following conjectures, indicate if they are true or false, explaining why.

   You can assume that functions $f(n)$ and $g(n)$ are asymptotically positive, i.e., they are positive from some point on ($\exists n_0 : f(n) > 0$ for all $n \geq n_0$)

   (a) $f(n) = O(g(n))$ implies that $g(n) = O(f(n))$

   (b) $f(n) = O(g(n))$ implies that $g(n) = \Omega(f(n))$

   (c) $f(n) + g(n) = \Theta(min(f(n), g(n)))$

   (d) $f(n) + g(n) = \Theta(max(f(n), g(n)))$

   (e) $(n + c)^k = \Theta(n^k)$, where c and k are positive integer constants

   (f) $f(n) + o(f(n)) = \Theta(f(n))$

   (g) $n^2 = \Theta(16^{\log_4 n})$

Growth Ratio

4. Imagine a program A running with time complexity $\Theta(f(n))$, taking t seconds for an input of size k. What would your estimation be for the execution time for an input of size $2k$ for the following functions: n, n^2 , n^3, 2^n, $\log_2 n$. Is this growth ratio constant for any k or is it changing?

5. Consider two programs implementing algorithms A and B, both trying to solve the same problem for an input of size n. They measured the execution times for test cases of different sizes and got the following table:

Algorithm	$n = 100$	$n = 200$	$n = 300$	$n = 400$	$n = 500$
A	0.003s	0.024s	0.081s	0.192s	0.375s
B	0.040s	0.160s	0.360s	0.640s	1.000s

   (a) Which program is more efficient? Why?

   (b) Could you produce a program that uses both algorithms in order to produce an algorithm C that would be at least as good as A and B for any test case?