Homework 2

2019.03.07

Note: When the exercise asks you to "design an algorithm for…," it always means that "designs an EFFICIENT algorithm for … and ANALYZES your algorithm". You should keep this in mind when writing solutions.

1. **Problem 3-2: *Relative asymptotic growths***

   Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, w, or Θ of B. Assume that $k \geq 1$, $\varepsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box, and **you should justify your answer**.

	A	B	O	o	Ω	w	Θ
a.	$\lg^k n$	n^ε					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\ln n}$					
f.	$\lg(n!)$	$\lg(n^n)$					

2. **Problem 3-3 Ordering by asymptotic growth rates**

   a. Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{30}$ of the functions satisfying $g_1 = \Omega(g_2)$, $g2 = \Omega(g_3)$, …, $g_{29} = \Omega(g_{30})$. Partition your list into equivalence classes such that functions $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(\, g(n)\,)$.
   About the definition of *, you can check the textbook.

$\lg(\lg^* n)$	$2^{\lg^* n}$	$(\sqrt{2})^{\lg n}$	n^2	$n!$	$(\lg n)!$
$(\frac{3}{2})^n$	n^3	$\lg^2 n$	$\lg(n!)$	2^{2^n}	$n^{1/\lg n}$
$\ln \ln n$	$\lg^* n$	$n \cdot 2^n$	$n^{\lg \lg n}$	$\ln n$	1
$2^{\lg n}$	$(\lg n)^{\lg n}$	e^n	$4^{\lg n}$	$(n+1)!$	$\sqrt{\lg n}$
$\lg^*(\lg n)$	$2^{\sqrt{2 \lg n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

   b. Give an example of a single nonnegative function $f(n)$ such that for all functions $g_i(n)$ in part (a), $f(n)$ is neither $O(g_i(n))$ nor $\Omega(g_i(n))$.

3. **Problem 3-4 Asymptotic notation properties**

Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures.

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

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

c. $f(n) = O(g(n))$ implies $\lg(f(n)) = O(\lg(g(n)))$, where $\lg(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.

d. $f(n) = O(g(n))$ implies $2^{f(n)} = O(2^{g(n)})$.

e. $f(n) = O((f(n))^2)$.

f. $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.

g. $f(n) = \Theta(f(n/2))$.

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

4. **Exercise 4.3-9:**

Solve the recurrence $T(n) = 2T(\sqrt{n}) + 1$ by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral.

5. **Exercise 4.2-1:**

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Use the substitution method to verify your answer.

6. **Problem 4-1: Recurrence examples**

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leqq 2$. Make your bounds as tight as possible, and justify your answers.

a. $T(n) = 2T(n/2) + n^3$

b. $T(n) = T(9n/10) + n$

c. $T(n) = 16T(n/4) + n^2$

d. $T(n) = 7T(n/3) + n^2$

e. $T(n) = 7T(n/2) + n^2$

f. $T(n) = 2T(n/4) + \sqrt{n}$

g. $T(n) = T(n-1) + n$

h. $T(n) = T(\sqrt{n}) + 1$

7. **Problem 4-3: More recurrence examples**

Give asymptotic upper and lower bounds for *T(n)* in each of the following recurrences. Assume that *T*(*n*) is constant for sufficiently small *n*. Make your bounds as tight as possible, and justify your answers.

a. $T(n) = 4T(n/3) + n \lg n$

b. $T(n) = 3T(n/3) + n / \lg n$

c. $T(n) = 4T(n/2) + n^2 \sqrt{n}$

d. $T(n) = 3T(n/3 - 2) + n / 2$

e. $T(n) = 2T(n/2) + n / \lg n$

f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

g. $T(n) = T(n-1) + 1/n$

h. $T(n) = T(n-1) + \lg n$

i. $T(n) = T(n-2) + 1/\lg n$

j. $T(n) = \sqrt{n}T(n-1) + n$