University of Houston

COSC 3320: Algorithms and Data Structures
Spring 2016

Homework 2

Due February 11, at the start of class

1. (a) Prove that the function $f(n) = na^{\log n}$, where a is a constant greater than 1, is $\Theta(n^c)$ for some constant c.
   (b) Prove that the function $f(n) = n^{1/\log n}$ is $O(1)$.
   (c) Prove that for any constant $a > 0$, $f(n) = \log n$ is $o(n^a)$.
   (d) Order the following functions by order of growth, that is, find an arrangement $f_1, f_2, \ldots, f_{20}$ of the functions such that $f_1 = O(f_2), f_2 = O(f_3), \ldots, f_{19} = O(f_{20})$. (Here $\log n$ means $\log_2 n$.)

n^2	$\dfrac{1}{\log n}$	$n^{4/5}$	1.5^n	$\dfrac{2^{\log n}}{2}$
$n \log\log n$	$\sqrt{\log n}$	$n^{\log_2 3}$	8	$\log\log\log n$
$\sqrt{n^5}$	$\log^{11/6} n$	$e^{\sqrt{n}}$	$\log\log n^3$	$\log n!$
$2^{\sqrt{\log n}}$	$\dfrac{n}{\log n}$	$\log\left(\dfrac{n}{\log n}\right)$	$\dfrac{\log n}{n}$	$n!$

2. Design recursive algorithms for the following problems:

   (a) Compute the n-th Fibonacci number F_n. Recall that the n-th Fibonacci number is defined as follows.

$$F_n = \begin{cases} 1 & \text{if } n = 0, 1, \\ F_{n-1} + F_{n-2} & \text{if } n \geq 2. \end{cases}$$

   (b) Compute the n-th power of a number x, x^n, with n non-negative integer. The algorithm should be designed in such a way that it is possible to write a recurrence relation for the total number of multiplications executed by the algorithm for which the Master Theorem applies. Write such a recurrence and apply the Master Theorem to obtain an asymptotic bound for it.

3. When possible, apply the Master Theorem to give asymptotic bounds for $T(n)$ for the following recurrences:

   (a)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 3T(n/3) + n/2 & \text{if } n > 1. \end{cases}$$

   (b)

$$T(n) = \begin{cases} 4 & \text{if } n = 1, \\ 4T(n/2) + 16n^{15/7} & \text{if } n > 1. \end{cases}$$

(c)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(n/2 + 2) + n^2 & \text{if } n > 1. \end{cases}$$

(d)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 4T(n/2) + n/\log n & \text{if } n > 1. \end{cases}$$

(e)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ \log n \cdot T(n/2) + n^2 & \text{if } n > 1. \end{cases}$$

(f)

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2T(n/2) + n/\log n & \text{if } n > 1. \end{cases}$$