CISC 5835—Algorithms for Data Analytics
Fall, 2019

Chapter 2 Homework

1. [10 points] Do Problem 2.3(a) in the text (page 71), but start with the recurrence $T(n) = 3T(n/2) + n$.

(a) [4 points] What is the general kth term in this case?

(b) [3 points] What value of k should be plugged in to get the answer?

(c) [3 points] What is the resulting solution of the recurrence? Do not use the Master Theorem, but use the answers to parts (a) and (b) of this question.

2. [20 points] Solve each of the following recurrences, giving a Θ-bound.

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

(b) $T(n) = 27T(n/3) + n^2$

(c) $T(n) = 9T(n/3) + n^2$

(d) $T(n) = T(n-1) + n^2$.

Hint: The Master Theorem doesn't apply here. Do a repeated substitution (as in Problem 1 above), maybe with a small value of n (say, $n = 5$) to see what's going on.

3. [10 points] You are given an infinite array $a[\cdot]$. You are *not* given the value of n. Describe an algorithm $\textsc{find}(x, a)$, where

- x is an integer,
- a is an infinite array, in which the first n cells contain integers in sorted order and the other cells are marked with ∞,

and whose return value is

$$\textsc{find}(x, a) = \begin{cases} j & \text{if } x = a[j] \text{ for some index } j, \\ \infty & \text{otherwise.} \end{cases}$$

The running time of this algorithm must be $O(\log n)$.

Hint: Note that you do not know the value of n. So, $\textsc{find}$'s first task is determining a good upper bound on the value of n.