Written Homework 06

Assigned: Wed 18 Nov 2015
Due: Wed 2 Dec 2015

Instructions:

- The assignment has to be uploaded to blackboard by the due date. NO assignment will be accepted after 11:59pm on that day.

- We expect that you will study with friends and often work out problem solutions together, but you must write up your own solutions, in your own words. Cheating will not be tolerated. Professors, TAs, and peer tutors will be available to answer questions but will not do your homework for you. One of our course goals is to teach you how to think on your own.

- We require that all homework submissions be neat, organized, and typeset. You may use plain text or a word processor like Microsoft Word or LaTeX for your submissions. If you need to draw any diagrams, however, you may draw them with your hand.

- To get full credit, show intermediate steps leading to your answers, throughout.

Problem 1 [20 pts (9,11)]: Recurrence Equations

i. Show that for every \( n \geq 1 \)
\[ \lceil n/3 \rceil + \lfloor 2n/3 \rfloor = n \]

ii. Consider the equation
\[
\begin{cases} 
T(1) = 1 \\
T(n) = T(\lceil n/3 \rceil) + T(\lfloor 2n/3 \rfloor) + 1 & \text{for } n > 1
\end{cases}
\]
Show by induction that \( T(n) \leq 2n - 1 \) for \( n \geq 1 \)

Problem 2 [21 pts, (10,11)]: Three Way Merge Sort

Consider a 3-MERGE-SORT algorithm where instead of dividing the list to be sorted into two sublists, one divides it into three. In other words, a) the original list \( L \) of length \( n \) is divided into three lists \( L_1, L_2, L_3 \) of length approximately \( n/3 \), b) each of the lists is recursively sorted by 3-MERGE-SORT, c) the sorted lists \( L_1, L_2, \) and \( L_3 \) are then merged into one grand sorted list \( L \).

i. Set up a recurrence equation for the running time \( T(n) \) of the above algorithm. In particular, explain how would you merge the three lists in step c), and how much time such merging would take.

ii. Solve the equation from i. How does the running time of 3-MERGE-SORT compare with that of the 2-way MERGE-SORT?
Problem 3 [39 pts; (13,13,13)]: Recurrence Equations

Follow the method described in the textbook (p. 181) to solve the following recurrences via iteration. You will need to establish a pattern for what the recurrence looks like after the k-th iteration (show your work). You need not formally prove (e.g., by induction) that your patterns are correct, though you will lose points if they are not. Your solutions may involve n raised to a power and/or logarithms of n. For example, a solution of the form $4^{\log_2 n}$ is unacceptable; this should be simplified as $(2^2)^{\log_2 n} = 2^{2\log_2 n} = (2^{\log_2 n})^2 = n^2$. Assume that $T(1) = 1$

i. $T(n) = 3T(n/4) + 2n^2$

ii. $T(n) = 3T(n/2) + n$

iii. $T(n) = T(n/9) + \sqrt{n}$

Problem 4 [10 pts (5,5)]: Growth Rates of Functions

Order the following functions according to their growth rate. That is, if function $g(n)$ appears after $f(n)$ in your list, we must have $g(n) \geq f(n)$ for all $n$ greater than some threshold value $n_0$.

i. Order the following functions according to their growth rate. That is, if function $g(n)$ appears after $f(n)$ in your list, we must have $g(n) \geq f(n)$ for all $n$ greater than some threshold value $n_0$.

$n^{\log_2 n}, \frac{3}{4} n^{1.5}, n\log_2 n, \sqrt{n}, \frac{1}{4} n^n, \log_2 n, n + 47, 2^n, \frac{1}{2} (\log_2 n)^n, \frac{1}{3} n!$

ii. Having the list of above functions sorted in the order of their asymptotic growth, for each consecutive pair $f(n) \prec g(n)$ give the minimal value $n_0 \geq 1$ such that for all $n \geq n_0$, $g(n) \geq f(n)$.

Problem 5 [10 pts (5 each)]: Sums

i. Evaluate the following sum. You must show your work, and your final answer should be a single integer.

$$\sum_{k=5}^{45} (6k + 5) =$$

ii. Derive a formula in terms of $a$ ($a \neq 0$) for the following sum. You must show your work, and your final formula should only contain $a$, and integers (but not $k$).

$$\sum_{k=5}^{45} a^{-k+2} =$$