CS140: Algorithms

Spring 2001

Assignment 3

This assignment is due at the start of class on Tuesday, Jan. 30. You may collaborate with your classmates to complete this assignment but you should reference your collaborators. You also must thoroughly understand any solutions you submit and must do your own write-up of those solutions. This assignment is worth 100 points (75 points if handwritten).

1. $\mathcal{A}$ is a divide and conquer algorithm with recurrence relation $T(n) = 4T(n/3) + n$. Answer the following questions about the work tree for $\mathcal{A}$ on input size $n = 3^m$.

   (a) What is the input size to a node at level i, where the root is at level 0?

   (b) How much work is done at a node at level i?

   (c) How many nodes are at level i?

   (d) What is the total amount of work done by the nodes at level i?

   (e) The root is at level 0. What level are the leaves at?

   (f) Write a series that represents the running time of the algorithm for inputs of size $n = 3^m$.

   (g) Solve the series.

   (h) Let the solution to the previous question be $f(n)$. At this point you've shown that the running time $T(n)$ satisfies the following: $T(n) = f(n)$ when n is a power of 3. Now we want to extend this result to arbitrary n.

      i. Show the $f(\lceil\lceil n\rceil\rceil) = O(f(n))$.

      ii. Use this result and the fact that $T(n)$ is monotonically increasing to argue that $T(n) = f(n)$ for all n.

2. Fill in the table with Y/N to the question $f(n) = \mathcal{X}(g(n))$ where $\mathcal{X} = O, \Omega, \Theta, o, \omega$. Assume that ϵ, k and c are constants satisfying the following: $\epsilon > 0$, $k > 1$, and $c > 1$.

$f(n)$	$g(n)$	O	Ω	Θ	o	ω
n^{ϵ}	$\log^k n$					
n^k	c^n					
$n^{\lg c}$	$c^{\lg n}$					
n^k	n^{2k}					
n^k	$(n/c)^k$					
c^n	c^{n+k}					

3. Give asymptotic upper and lower bounds for the following recurrences. Your bounds should be as tight as possible. You may assume that $T(n)$ is constant for $n <= 2$.

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

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

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

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

   (e) $T(n) = \sqrt{n}T(\sqrt{n}) + n$

4. Construct a heap on the set $\{3, 7, 9, 1, 2\}$; show the heap as a binary tree as well as an array. Show the heap after each of the following successive operations: Insert 4, Extract-min, Insert 1.