CS140: Algorithms

Spring 2001

Solutions to Assignment 3

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?
   $n/3^i$

   (b) How much work is done at a node at level i?
   $n/3^i$

   (c) How many nodes are at level i?
   4^i

   (d) What is the total amount of work done by the nodes at level i?
   $n(\frac{4}{3})^i$

   (e) The root is at level 0. What level are the leaves at?
   $m = \log_3 n$

   (f) Write a series that represents the running time of the algorithm for inputs of size $n = 3^m$.
   $\sum_{i=0^m} n(\frac{4}{3})^i$

   (g) Solve the series.
   $\sum_{i=0^m} cn(4/3)^i = \frac{4}{9}n^{\log_3 4} - \frac{1}{3}n$

   (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 $f(\lceil\lceil n\rceil\rceil) = O(f(n))$.
      $f(\lceil\lceil n\rceil\rceil) = \frac{4}{9}(\lceil\lceil n\rceil\rceil)^{\log_3 4} - \frac{1}{3}\lceil\lceil n\rceil\rceil \leq (2n)^{\log_3 4} = O(n^{\log_3 4})$.

      ii. Use this result and the fact that $T(n)$ is monotonically increasing to argue that $T(n) = f(n)$ for all n.
      Since $T(n)$ is monotonically increasing $T(n) \leq T(\lceil\lceil n\rceil\rceil_3)$, where $\lceil\lceil n\rceil\rceil_n$ is the smallest power of 3 that is at least n. By the obvious extension of the previous proof $f(\lceil\lceil n\rceil\rceil_3) = O(f(n))$.

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	Y	N	N	Y
n^k	c^n	Y	N	N	Y	N
$n^{\lg c}$	$c^{\lg n}$	Y	Y	Y	N	N
n^k	n^{2k}	Y	N	N	Y	N
n^k	$(n/c)^k$	Y	Y	Y	N	N
c^n	c^{n+k}	Y	Y	Y	N	N

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) $T(n) = T(n/3) + n = \theta(n)$

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

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

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

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

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.