CS140: Algorithms

Fall 2000

Solutions to Assignment 3

1. Algorithm Resquare is a recursive algorithm that works on inputs of size $n = 2^{2^m}, m \geq 0$. On input of size 2 the algorithm performs a constant amount of work. On input size $n = 2^{2^m}, m > 0$, the algorithm makes $\sqrt{n}$ recursive calls, each on an instance of size $\sqrt{n}$. The algorithm performs an additional cn steps to combine the results of the recursive calls. Draw the work tree for Resquare for $m = 0, m = 1$, and $m = 2$. Answer the following questions with respect to the work tree for Resquare for arbitrary m.

   (a) How many nodes are at level i, where the root is at level 0?
   $2^{2^m - 2^{m-i}}$.

   (b) What is the instance size associated with a node at level i?
   $2^{2^{m-i}}$.

   (c) How much work is associated with a node at level i?
   $c \cdot 2^{2^{m-i}}$

   (d) How many levels are in the work tree?
   $m + 1$

   (e) What is the running time of Resquare?
   $\sum_{i=0}^{m} c \cdot 2^{2^{m-i}} \cdot 2^{2^m - 2^{m-i}} = c \sum_{i=0}^{m} 2^{2^m} = c(m+1)2^{2^m} = O(n \lg \lg n)$.

2. Complete the following table giving asymptotic upper and lower bounds for the recurrence relations shown. To receive full credit your bounds should be as tight as possible.

Recurrence	Upper Bound	Lower Bound
$T(n) = 2T(n/2) + n^3$	$O(n^3)$	$\Omega(n^3)$
$T(n) = 3T(n/2) + n^3$	$O(n^3)$	$\Omega(n^3)$
$T(n) = 7T(n/3) + n$	$O(n^{\log_3 7})$	$\Omega(n^{\log_3 7})$
$T(n) = 2T(n/4) + n$	$O(n)$	$\Omega(n)$
$T(n) = 2T(n/4) + \sqrt{n}$	$O(\sqrt{n} \lg n)$	$\Omega(\sqrt{n} \lg n)$
$T(n) = 2T(n/4) + \lg n$	$O(\sqrt{n})$	$\Omega(\sqrt{n})$
$T(n) = T(n-1) + n$	$O(n^2)$	$\Omega(n^2)$
$T(n) = T(\sqrt{n}) + 1$	$O(\lg \lg n)$	$\Omega(\lg \lg n)$