CS140: Algorithms

Fall 2001

Assignment 2

This assignment is due at the start of class on Monday, Sept. 17. You may collaborate in small groups 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. You may not refer to solutions to assignments given in previous semesters of CS140. This assignment is worth 100 points (75 points if handwritten).

1. Order the following functions by increasing rate of growth. Indicate any equivalence classes you find.

$\lg^* n$	$\lg(\lg^* n)$	$2 \uparrow\uparrow (\lg^*(n))$	$2^{\lg^*(n)}$	$(\lg n)!$
$\lg(n!)$	n^2	$2n + \lg n$	$(\lg n)^{\lg n}$	$2^{n/2}$
$\log\log n$	$2^{\lg n}$	$(\frac{3}{2})^n$	$n^{1/\lg n}$	5
$\lg(n^{\lg n})$	$(\frac{1}{2})^n$	$\sqrt{n}$	$2^{3\uparrow\uparrow n}$	4^n

Ans:

(a) $(\frac{1}{2})^n$

(b) $5, n^{1/\lg n}$

(c) $\lg(\lg^* n)$

(d) $\lg^* n$

(e) $2^{\lg^*(n)}$

(f) $\log\log n$

(g) $\sqrt{n}$

(h) $2n + \lg n,\ \ 2^{\lg n},\ \ 2 \uparrow\uparrow (\lg^*(n))$

(i) $\lg(n!)$

(j) n^2

(k) $(lgn)!$

(l) $(\lg n)^{\lg n}$

(m) $2^{n/2}$

(n) $(\frac{3}{2})^n$

(o) 4^n

(p) $2^{3\uparrow\uparrow n}$

Note: functions in the same equivalence class are shown on the same line.

2. $\mathcal{A}$ 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. $\sum_{i=0^m} n(\frac{4}{3})^i = \frac{4}{9}n^{\log_3 4} - \frac{1}{3}n$

   (h) Now prove that the big-O bound you obtain through the previous analysis holds for all values of 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 $T(n) \leq T(\lceil\lceil n \rceil\rceil_3)$, whe re $\lceil\lceil n \rceil\rceil_n$ is the smallest power of 3 that is at least n. By the obvious extension of the previous proof $f(\lceil\lceil n \rceil\rceil_3) = O(f(n))$.

3. Answer the work tree questions posed in the previous problem for each of the recurrence shown below. You may assume that $T(n)$ is constant for $n <= 2$.

   (a) $T(n) = T(n/3) + n$ (Assume $n = 3^m$ for some m.)

      i. Input size at level i: $3/3^i$
      ii. Work done by a node at level i: $3/n^i$
      iii. Number of nodes at level i:1
      iv. Total work done by nodes at level i: $3/n^i$
      v. Leaf level: m
      vi. Summation: $\sum_{i=0}^{m} n/3^i$
      vii. Solution: $T(n) = O(n)$

(b) $T(n) = 7T(n/3) + n^2$ (Assume $n = 3^m$ for some m.)

i. Input size at level i: $n/3^i$
ii. Work done at a node in level i: $n^2/3^{2i} = n^2/9^i$
iii. Number of nodes at level i: 7^i
iv. Total work done by nodes at level i: $n^2(7/9)^i$
v. Leaf level: m
vi. Summation: $\sum_{i=0}^{m} n^2(7/9)^i$
vii. Solution: $T(n) = O(n^2)$

(c) $T(n) = 2T(n/4) + \sqrt{n}$ (assume $n = 4^m$ for some m.)

i. Input size at level i: $n/4^i$
ii. Work done by a node at level i: $(n/4^i)^{1/2} = \sqrt{n}/2^i$
iii. Number of nodes at level i: 2^i
iv. Total work done by nodes at level i: $\sqrt{n}$
v. Leaf level: m
vi. Summation: $\sum_{i=0}^{m} \sqrt{n}$
vii. Solution: $T(n) = O(\sqrt{n} \lg n)$

(d) $T(n) = T(\sqrt{n}) + 1$ (Assume $n = 2^{2^m}$ for some m.)

i. Input size at level i: $2^{2^{m-i}}$
ii. Work done by a node at level i: 1
iii. Number of nodes at level i: 1
iv. Total work done by nodes at level i: 1
v. Leaf level: m (Assume we stop when $n = 2$.)
vi. Summation: $\sum_{i=0}^{m} 1$
vii. Solution: $T(n) = O(\lg \lg n)$

(e) $T(n) = \sqrt{n}T(\sqrt{n}) + n$ (Assume $n = 2^{2^m}$ for some m.)

i. Input size at level i: $2^{2^{m-i}}$
ii. Work done a node at level i: $2^{2^{m-i}}$
iii. Number of nodes at level i: $2^{2^m}/2^{2^{m-i}}$
iv. Total work done at level i: 2^{2^m}
v. Leaf level: m
vi. Summation: $\sum_{i=0}^{m} 2^{2^m}$
vii. Solutions $T(n) = O(n \lg \lg m)$