CS140: Algorithms

Fall 2000

Practice Exam 1

This is for your own benefit and is not to be turned in for grading.

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

$$\log_3 n^3,\ \ n^2 + \log n,\ \ n^2 \lg n,\ \ \lg^2 n,\ \ T*(n),\ \ T*(2n)$$

2. For each of the following pair $f(n)$ and $g(n)$ state whether $f(n) = O(g(n))$ and/or $g(n) = O(f(n))$.

   (a) $f(n) = n^2 + n$ and $g(n) = n^3 - n^2$

   (b) $f(n) = n^2$ and $g(n) = f(2n)$

   (c) $f(n) = n!$ and $g(n) = f(2\lceil n/2 \rceil)$

3. Algorithm $\mathcal{A}$ works on inputs of size $n = T^*(m)$, $m = 0, 1, ...$; i.e. towers of 2. On an input of $n = 1$ the algorithm performs a constant number of steps. On an input of size $T^*(m)$, $m > 0$, $\mathcal{A}$ makes $n/lg(n)$ recursive calls, each on an instance of size $\lg(n)$. It then performs an additional cn steps to produce its output. Consider the work tree for $\mathcal{A}$ on input of size $n = T^*(m)$.

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

   (b) How many nodes are at level i?

   (c) What is the total work done by the nodes at level i?

   (d) How many levels are in the work tree?

   (e) What is the running time of the algorithm?

   (Note: Recall the notation for the ith iterated log of n is $lg^{(i)}n$; i.e. $lg^{(0)}n = n$ and for $i > 0$, $lg^{(i)}n = lg(lg^{(i-1)}n)$. This may be usefule.)

4. The set $S = 3, 6, 2, 1, 8, 9, 4, -1$ is input to sorting algorithm $\mathcal{A}$. What is the first swap made if $\mathcal{A}$ is

   (a) Bubble-sort

   (b) Insertion-sort

   (c) Merge-sort

5. Consider the Find-Gap-5 problem defined as follows:

Input: A list S of $n > 2$ distinct integers sorted in ascending order.

Output: An index i such that $S[i] - S[i-1] \geq 5$ if such an index exists, and otherwise a message stating "No 5-gap exists."

An algorithm for the problem has random access to the input and can choose to read (or not) any element of the list. In other words, the algorithm makes queries of the form "What is S[i]?".

Let $T(n)$ be the worst-case number of elements that must be read. Give the best upper and lower bounds you can for $T(n)$. Your bound should be exact, not in big-O notation. You can assume that n is a power of 2 if you find it helpful.