Computer Science and Engineering 373

Data Structures and Algorithms

Steve Tanimoto, Instructor
Autumn Quarter, 2006

Follow-up Problems for the Midterm

Instructions Do as many of these problems as you wish. For each problem you do, if you turn in a complete and correct solution, you will be credited with up to half the points you missed on the corresponding problem in Part 2 of the midterm. In cases where you missed k points and k is odd, you can make up as many as $\lfloor k/2 \rfloor$ points.

1. The fibonacci numbers are defined by

$$f_1 = 1,\ f_2 = 1,\ \text{and for } n > 2,\ f_n = f_{n-1} + f_{n-2}$$

   Prove by induction that for all $n \geq 2$,

$$(f_n)^2 = f_{n-1} \cdot f_{n+1} - (-1)^n$$

   Clearly mark your basis, induction hypothesis, and induction step.

2. Using the formal definitions of Big Theta and Big O relationships, show that

$$f(n) \text{ is in } \Theta(g(n))$$

   where $f(n) = (n + 10)(n + 2)$ and $g(n) = 5n^2 100n + 50$. (Give actual values of c and n_0 as necessary.)

3. Construct the BST whose postorder traversal gives us the sequence of keys A, C, B, G, F, H, E, D.

4. Using pseudocode, give an algorithm that prints the data of a binary tree in a preorder traversal.

5. A certain sorting algorithm works as follows. It starts with a parameter s set to 1. It takes the list of n integer keys and puts them in a queue Q1. Then, it dequeues them one at a time and computes $p = \lfloor (k/s) \rfloor \bmod 2$. Here k is the key. If $p = 0$, it puts k in queue Q2. If $p = 1$, it puts k in Q3. When Q1 is empty, it dequeues all elements of Q2 enqueueing them into Q1 and then it dequeues all elements of Q3 enqueueing them, too, into Q1. Next, it doubles s and does this all again. It keeps doubling s and doing the enqueuing and dequeuing until s is greater than or equal to n. After the last step of dequeueing all elements of Q2 and Q3, the sorted list is in Q1.

   Determine a formula for the running time $T(n)$ of this algorithm. Then determine whether $T(n)$ is in $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^2)$, etc.