3) (10 pts) ANL (Recurrence Relations)

Use the iteration technique to solve the following recurrence relation in terms of n:

$$T(n) = 2T(n/2) + 1, for\ all\ integers\ n > 1$$
$$T(1) = 1$$

Find a tight Big-Oh answer.

$$T(n) = 2T\left(\frac{n}{2}\right) + 1$$
$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + 1$$
$$T(n) = 2\left(2T\left(\frac{n}{4}\right) + 1\right) + 1$$
$$T(n) = 4T\left(\frac{n}{4}\right) + 2 + 1$$
$$T(n) = 4T\left(\frac{n}{4}\right) + 3$$
$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + 1$$
$$T(n) = 4\left(2T(\frac{n}{8}) + 1\right) + 3$$
$$T(n) = 8T(\frac{n}{8}) + 4 + 3$$
$$T(n) = 8T(\frac{n}{8}) + 7$$

Based on these three iterations, we see that after k iterations, the recurrence is

$$T(n) = 2^kT(\frac{n}{2^k}) + (2^k - 1)$$

Plug in the value of k such that $\frac{n}{2^k} = 1$ to this recurrence. This means that $2^k = n$. Substituting, we get:

$$T(n) = nT(1) + (n - 1)$$
$$T(n) = n + (n - 1)$$
$$T(n) = 2n - 1$$

It follows that T(n) = O(n).

Grading: 2 pts for iteration with T(n/4), 2 pts for T(n/8). 2 pts for general expression after k iterations, 1 pt for the value to plug in for k. 3 pts to finish the problem.

Computer Science Foundation Exam

January 11, 2020

Section II B

ALGORITHMS AND ANALYSIS TOOLS

SOLUTION

NO books, notes, or calculators may be used,
and you must work entirely on your own.

Question #	Max Pts	Category	Score
1	10	DSN	
2	10	DSN	
3	5	ALG	
TOTAL	25		

You must do all 3 problems in this section of the exam.

Problems will be graded based on the completeness of the solution steps and not graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all be neat. For each coding question, assume that all of the necessary includes (stdlib, stdio, math, string) for that particular question have been made.