3) (10 pts) ANL (Recurrence Relations)

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

$$T(n) = 2T(n-1) + 2^n, for\ all\ integers\ n > 0$$
$$T(0) = 1$$

Please give an **<u>exact closed-form answer in terms of n,</u>** instead of a Big-Oh answer.

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

$$= 2(2T(n-2) + 2^{n-1}) + 2^n$$

$$= 4T(n-2) + 2^n + 2^n$$

$$= 4T(n-2) + 2(2^n)$$

$$= 4(2T(n-3) + 2^{n-2}) + 2(2^n)$$

$$= 8T(n-3) + 2^n + 2(2^n)$$

$$= 8T(n-3) + 3(2^n)$$

After k steps, we have: $= 2^k T(n-k) + k(2^n)$

Let k = n, then we have that $T(n) = 2^n T(n-n) + n(2^n)$

$$= 2^n T(0) + n(2^n)$$

$$= 2^n + n(2^n)$$

$$= (n+1)(2^n)$$

Grading: 2 pts for iteration with T(n-2), 2 pts for iteration with T(n-3), 2 pts for general guess after k steps. 1 pt for plugging in k = n (or k = n-1), 3 pts for simplifying that to the final answer.

Computer Science Foundation Exam

August 31, 2019

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	5	ALG	
3	10	DSN	
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.