3) (10 pts) ANL (Recurrence Relations)

Use the iteration technique to find **an exact closed-form** solution to the recurrence relation defined below for all positive integers n:

$$T(1) = 1$$
$$T(n) = 2T(n-1) + 5, for\ all\ integers\ n \geq 2$$

Please explicitly show the work for the first three iterations before attempting to find the form for an arbitrary iteration, followed by arriving at the closed form. Hint: Your answer should be of the form $T(n) = a(b^n) + c$, where *a*, *b*, and *c* are all integers.

Here are the first three iterations:

$T(n) = 2T(n-1) + 5$ // Iteration #1 **Grading: 1 pt**
$T(n) = 2(2T(n-2) + 5) + 5$
$T(n) = 4T(n-2) + (10 + 5)$
$T(n) = 4T(n-2) + 15$ // Iteration #2 **Grading: 1 pt**
$T(n) = 4(2T(n-3) + 5) + 15$
$T(n) = 8T(n-3) + (20 + 15)$
$T(n) = 8T(n-3) + 35$ // Iteration #3 **Grading: 2 pts**

After k iterations, we have:

$T(n) = 2^k T(n-k) + 5(2^k - 1)$ **Grading: 2 pts**

Since we know T(1), plug in k = n – 1 into this formula: **Grading: 1 pt**

$T(n) = 2^{n-1}T(n-(n-1)) + 5\left(2^{(n-1)} - 1\right)$
$= 2^{n-1}T(1) + 5(2^{n-1}) - 5$ **Grading: 1 pt**
$= 2^{n-1} + 5(2^{n-1}) - 5$
$= 6(2^{n-1}) - 5$ **Grading: 1 pt**
$= (3)(2)(2^{n-1}) - 5$
$= 3(2^n) - 5$ **Grading: 1 pt**

Note: 1 pt is allocated to factor out the 2 from the 6 and include it in the exponent.

Computer Science Foundation Exam

January 11, 2025

Section D

ALGORITHMS

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

SOLUTION

Question #	Max Pts	Category	Score
1	10	DSN	
2	10	ALG	
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.