Recurrence Relations

On this worksheet, we will only count 'basic units': assignments statements (A) and comparisons (C).

1) Given the recursive factorial algorithm (a) write the recurrence equation, (b) solve the recurrence equation by repeated substitution

Algorithm factorial(n):
    Input: An integer $n \geq 1$.
    Output: $n!$.

    if $n \leq 1$ **then**
        return 1
    return $n * \text{factorial}(n-1)$

$$T(n) = \begin{cases} 2, & n = 1 \\ T(n-1) + 2, & n \geq 2 \end{cases}$$

$T(n) = T(n-1) + 2$
$T(n-1) = T(n-2) + 2$
$T(n-2) = T(n-3) + 2$
...
$T(2) = T(1) + 2$
$T(1) = 2$

Method 1: (bottom-up):
$T(1) = 2$
$T(2) = T(1) + 2 = (2) + 2 = 4$
$T(3) = T(2) + 2 = (4) + 2 = 6$
$T(4) = T(3) + 2 = (6) + 2 = 8$
...
$T(n) = 2n$

Goal: to characterize a recurrence equation where no references to the function T appear on the right-hand side

Method 2: (top-down):
$T(n) = T(n-1) + 2$
$T(n) = (T(n-2) + 2) + 2 = T(n-2) + 2(2)$
$T(n) = (T(n-3) + 2) + 2 + 2 = T(n-3) + 3(2)$
...
$T(n) = T(n-i) + i(2) = T(n-i) + 2i$

This general form shifts to the base case when $i = n-1$
$T(n) = T(n-(n-1)) + 2(n-1)$
$T(n) = T(1) + 2n - 2$
$T(n) = 2n$

2) Given the recursive arrayMax algorithm (a) write the recurrence equation, (b) solve the recurrence equation by repeated substitution

Algorithm hanoiRecursive(n, A, B, C):
    Input: An integer $n \geq 1$, pegs A, B, C
    Output: n disks from A to (B or C) in minimum moves

    if $n = 1$ **then**
        move(A,C)
    else
        hanoiRecursive(($n-1$), A, C, B)
        move(A,C)
        hanoiRecursive(($n-1$), B, A, C)
    end

$$T(n) = \begin{cases} 1, & n = 1 \\ 2T(n-1) + 1, & n \geq 2 \end{cases}$$

$T(n) = 2T(n-1) + 1$
$T(n-1) = 2T(n-2) + 1$
$T(n-2) = 2T(n-3) + 1$
...
$T(2) = 2T(1) + 1$
$T(1) = 1$

Method 1: (bottom-up)
$T(1) = 1$
$T(2) = 2(1) + 1 = 3$
$T(3) = 2(3) + 1 = 7$
$T(4) = 2(7) + 1 = 15$
...
$T(n) = 2^n - 1$

Method 2 (top-down):
$T(n) = 2T(n-1) + 1$
$T(n) = 2(2T(n-2) + 1) + 1 = 2^2T(n-1) + 2 + 1$
$T(n) = 2(2(2T(n-3) + 1) + 1) + 1$
$= 2^3T(n-3) + 4 + 2 + 1$
$= 2^3T(n-3) + 2^2 + 2^1 + 2^0$
...
$T(n) = 2^iT(n-i) + 2^{i-1} + 2^{i-2} + \cdots + 2^2 + 2^1 + 2^0$

This general form shifts to the base case when $i = n-1$
$T(n) = 2^{n-1}T(1) + 2^{n-2} + 2^{n-3} + \cdots + +2^2 + 2^1 + 2^0$
$T(n) = 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + +2^2 + 2^1 + 2^0$
$T(n) = 2^n - 1$