CS330 – Order of Growth with Recursion

Purpose: Counting the number of times things get executed with recursive algorithms can sometimes be more challenging than with loops. We will examine how recurrence relations describe this running time and learn how to solve them for a non-recursive solution.

Knowledge: This activity will help you become familiar with the following content knowledge:

- Solving recurrence relations.
- Determining order of growth for recursive algorithms

Activity: With your group perform the following tasks and answer the questions. You will be reporting your answers back to the class in 60 minutes.

1. Consider this recursive function for computing the length of a list:

```
int length(ListNode list)
 {
   if (0 == list) return 0;
   else return 1+length(list.getNext());
 }
```

If the list is empty the running time will be 0. If the list is of length $n > 0$, the running time will be 1 plus the running time for the list of length $n - 1$.
So the function $T(n)$ which returns the running time of a list of length n is:

$T(0) = 0$
$T(n) = T(n-1) + 1$

A "closed form" of the function $T(n)$, is one that does not contain any recursion. Can you give an educated guess for the closed form solution for $T(n)$?

2. We want to do better than just guessing at a closed form solution. We can start by using a technique called repeated substitution to see a pattern.

   Let's start with the recurrence relation:

   $T(1) = a$
   $T(n) = T(n-1) + b$

   We will do the substitution of the recursion a few times to see a pattern:

$$\begin{aligned} T(n) &= T(n-1) + b \\ &= T(n-2) + b + b = T(n-2) + 2b \\ &= T(n-3) + 3b \\ &\ldots \\ &= T(n-i) + ib \end{aligned}$$

   What value of i will get us to the base case of $T(1)$?

   Plug in that value of i and solve with the base case value to get the closed form solution.

3. Finally we should check our possible solution by using mathematical induction. Your solution above should be

$$T(n) = a + (n-1)b$$

   Does this closed form solution work for the base case? Why?

   If our closed form solution works for $T(n-1)$ then does our possible closed form solution give us the correct value for $T(n)$? Why?

4. Consider this recurrence relation:

   $T(1) = 1$
   $T(n) = 2T(n/2) + 1$

   We will do the substitution of the recursion a few times to see a pattern:

$$\begin{aligned} T(n) &= 2T(n/2) + 1 \\ &= 2^2T(n/2^2) + 2 + 1 \\ &= 2^3T(n/2^3) + 2^2 + 2^1 + 2^0 \\ &\ldots \\ &= 2^iT(n/2^i) + 2^{i-1} + \ldots + 2^1 + 2^0 \end{aligned}$$

   What value of i will get us to the base case of $T(1)$?

5. $T(1) = 1$
   $T(n) = 2T(n/2) + 1$

$$\begin{aligned} T(n) &= 2T(n/2) + 1 \\ &= 2^2T(n/2^2) + 2 + 1 \\ &= 2^3T(n/2^3) + 2^2 + 2^1 + 2^0 \\ ... \\ &= 2^iT(n/2^i) + 2^{i-1} + ... + 2^1 + 2^0 \end{aligned}$$

   Given that an i of $\log_2 n$ gives us the base case of $T(1)$, what is the closed form solution of $T(n)$?

   Hint: $\sum_{i=0}^{N} 2^i = 2^{N+1} - 1$

   Be sure to check out your possible solution with induction!

Complete the following assignments to be submitted for grading. Each should be done individually but you can consult with a classmate to discuss your strategies.

Assignment 1:
For each of the following recurrence relations, get a closed form using repeated substitution and then verify your answer with mathematical induction. Finally, give the order of growth for each.

1. $T(n) = T(n-1) + 4, T(1) = 2$
2. $T(n) = T(n-1) + n, T(1) = 1$
3. $T(n) = 2T(n/2) + n, T(1) = 1$

Criteria for Success: You have both a closed form solution and a proof of its correctness with mathematical induction for each recurrence relation. Additionally you have the order of growth for each.

Assignment 2:
I will give you a shortcut for solving recurrence relations like the previous problem called the Master Theorem.
Suppose $T(n) = aT(n/b) + f(n)$ where $f(n) = \Theta(n^d)$ with $d \geq 0$. Then $T(n)$ is:

- $\Theta(n^d)$ if $a < b^d$
- $\Theta(n^d \lg n)$ if $a = b^d$
- $\Theta(n^{\log_b(a)})$ if $a > b^d$

Use the Master Theorem to solve the last recurrence relation in the previous problem, explaining what the values are for a, b and d, as well as the order of growth.

Criteria for Success: You clearly explain how the last recurrence relation can be solved by the Master Theorem and give the order of growth.