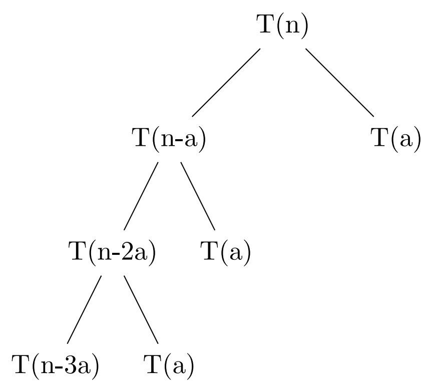
Caio Brighenti
COSC 302 - Analysis of Algorithms – Spring 2019
Lab 5

9. Recursion tree method.
The recurrence is $T(n) = T(n-a) + T(a) + cn$ for all $a \geq 1$ and $c > 0$. Thus, the recursion tree is:



Let the first level of the tree ($T(n)$) be $i = 0$, and each subsequent level sum one to i. Then, we see that the leftmost leaf in each level is equal to $T(n - ia)$. We assume the base case of the algorithm happens at 0, thus lowest level of the tree will be $T(1)$. Using this, we can solve for the total number of levels, i, as follows.

$$n - ia = 1 \tag{1}$$
$$ia = n - 1 \tag{2}$$
$$i = \frac{n-1}{a} \tag{3}$$

Thus, the total number of levels i is $\frac{n-1}{a}$. We must now find the running time per call. We have this information from the recurrence. Each call takes $T(a) + cn$ time, but as the linear term dominates, then the running time is $\Theta(n)$. Thus, the solution to the recurrence is as follows:

$$(\frac{n-1}{a})(cn) = \Theta(n)\Theta(n) \in \Theta(n^2)$$

As we have a factor n multiplied by another factor n. Each other element is a constant.