Homework 3
CS 3385

1. Show that $T(n) = T(n-1) + 1$ is $O(n)$ using the substitution method.
2. Show that $T(n) = 2T(\lfloor n/2 \rfloor) + n + 1$ is $O(n \lg n)$ using the substitution method.
3. Show that $T(n) = 2T(n-1) + n$ is $O(n^2)$ using the substitution method.
4. Show that $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$ using the substitution method.
5. Show that $T(n) = 2T(\lfloor n/2 \rfloor + 1) + n$ is $O(n \lg n)$ using the substitution method.
6. Show that $T(n) = T(n-1) + n$ is $\Omega(n^2)$ using the substitution method. *Hint*: Show that $cn^2 \leq T(n)$ for some c and $n \geq n_0$. You may find it easier to show that $T(n) \geq cn^2$.
7. Sam Smartypants likes how splitting the problem up into halves in merge sort reduces the sorting problem from $O(n^2)$ to $O(n \lg n)$. He decides that splitting the array into thirds will make things even better. That is, he decides to make a recursive call on each third of the array and then merge them.
   (a) Assuming that n is a power of three, that $T(1) = 1$, and that the running time of the merge step is exactly n, give a recurrence for the running time of Sam's algorithm.
   (b) Find the solution to the recurrence in big-O notation and prove it using the substitution method.