Michael Sanchez
I pledge my honor that I have abided by the Stevens Honor System.
4. Consider the following algorithm.

ALGORITHM Mystery(n)
//Input: A nonnegative integer n
S ← 0
for i ← 1 to n do
    S ← S + i ∗ i
return S

a. What does this algorithm compute?
The sum of all perfect squares from 1 to n
b. What is its basic operation?
Reassignment
c. How many times is the basic operation executed?
The basic operation is executed n times
d. What is the efficiency class of this algorithm?
$\Theta(n)$
e. Suggest an improvement, or a better algorithm altogether, and indicate its efficiency class. If you cannot do it, try to prove that, in fact, it cannot be done.

1. Solve the following recurrence relations.
a. $x(n) = x(n - 1) + 5$ for $n > 1$, $x(1) = 0$
$x(n) = 5n-5$
b. $x(n) = 3x(n - 1)$ for $n > 1$, $x(1) = 4$
$x(n) = 4*3^n$
c. $x(n) = x(n - 1) + n$ for $n > 0$, $x(0) = 0$
$x(n)=n(n-1)/2$
d. $x(n) = x(n/2) + n$ for $n > 1$, $x(1) = 1$ (solve for $n = 2k$)
$x(n) = 2n-1$
e. $x(n) = x(n/3) + 1$ for $n > 1$, $x(1) = 1$ (solve for $n = 3k$)

3. Consider the following recursive algorithm for computing the sum of the first n cubes: S(n) = 13 + 23 + ... + n3.

ALGORITHM S(n)
//Input: A positive integer n
//Output: The sum of the first n cubes
if n = 1 return 1
else return S(n − 1) + n ∗ n ∗ n

a. Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.
$\forall n$: $\{n > 1: S(n) = S(n-1) + n^3, S(1) = 1\}$
$S(n)=n^2(n+1)^2/2$
b. How does this algorithm compare with the straightforward nonrecursive algorithm for computing this sum?
This algorithm has a the same time complexity as the straightforward nonrecursive algorithm, $\Theta(n)$ if using a loop, but a time complexity of $\Theta(1)$ if just using the equation above