CST 370
Homework (Complexity Analysis)

1. 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?

b. What is its basic operation?

c. How many times is the basic operation executed?

d. What is the efficiency class of this algorithm?

2. Consider the following algorithm.

ALGORITHM *Secret*(A[0…n-1])

```
//Input: An array A[O..n - 1] of n real numbers
minval ←A[O]; maxval ←A[O]
for i ←1to n- 1 do
        if (A[i] < minval)
                minval ←A[i]
        if (A[i] > maxval)
                maxval ←A[i]
return maxval - minval
```

a. What does this algorithm compute?

b. What is its basic operation?

c. How many times is the basic operation executed?

d. What is the efficiency class of this algorithm?

3. Compute the following sums.

a. $1 + 3 + 5 + 7\ldots. 999$

b. $2 + 4 + 8 + 16 + \ldots + 1024$

4. Climbing stairs Problem: Find the number of different ways to climb an n-stair stair- case if each step is either one or two stairs.

5. Solve the following recurrence relations.

a. $x(n) = x(n-1) + 5$ for $n > 1$, $x(1) = 0$

b. $x(n) = 3x(n-1)$ for $n > 1$, $x(1) = 4$

c. $x(n) = x(n-1) + n$ for $n > 0$, $x(0) = 0$