**CS 477/677 Analysis of Algorithms**

**Homework 2**

**Due September 20, 2015**

**1. (U & G-required) [30 points]** Consider the following algorithm.

```
ALGORITHM Enigma(A[0..n − 1])
//Input: An array A[0..n − 1, 0..n − 1] of integer numbers
for i ← 0 to n − 2 do
        for j ← i +1 to n − 1 do
                if A[i] = A[j]
                        return false
return true
```

a) [5 points] What does this algorithm do?

b) [25 points] Compute the running time of this algorithm.

**2. (U & G-required) [30 points]**

Solve the following recurrences using the method of your choice.

a) [15 points] $T(n) = 7T\left(\frac{n}{2}\right) + n^2$

b) [15 points] $T(n) = T\left(\frac{9n}{10}\right) + n$

**3. (U & G-required) [40 points]**

Consider the following recursive algorithm for computing the sum of the first n cubes:

S(n) $= \mathbf{1^3 + 2^3 + \cdots + n^3}$

```
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) [20 points] Write and solve a recurrence relation for the number of multiplications made by this algorithm and solve it.

b) [20 points] How does this algorithm compare with the straightforward non-recursive algorithm for computing this function?

**4. (G-Required) [20 points]**

Consider the following recursive algorithm:

**ALGORITHM** *Q* **(n)**

// Input: A positive integer n

**if** *n =1*

    **return** *1*

**else**

    **return** *Q(n-1) + 2n -1*

a) [10 points] Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

b) [10 points] Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

**Extra credit**

**5. [20 points]** Solve the following recurrence using the method of your choice:

$T(n) = T\left(\sqrt{n}\right) + 1$

Hint: think iteration.