1. (U & G-required) [20 points] Consider the following algorithm.

```plaintext
ALGORITHM Enigma(A[0..n-1])
//Input: An array A[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) [15 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) \[ T(n) = 2T\left(\frac{n}{2}\right) + n^3 \]

(b) \[ T(n) = T\left(\sqrt{n}\right) + 1 \] (Hint: use the same substitution we made in the example in class)

(c) \[ T(n) = 3T\left(\frac{n}{2}\right) + n \lg n \]

3. (U & G-required) [50 points] (U-required)

(a) [25 points] Draw the recursion tree for \( T(n) = 4T(n/2) + n \) and provide a tight asymptotic bound on its solution.

(b) [25 points] Use the substitution method to show that the solution to the recurrence

\[ T(n) = 2T(n/2 + 17) + n \]

is \( O(n \lg n) \).
4. (G-Required) [20 points] Use a loop invariant to prove that the following algorithm computes $a^n$:

\[
\text{Exp}(a, n) \\
\begin{align*}
i &\leftarrow 1 \\
pow &\leftarrow 1 \\
\text{while } (i \leq n) \\
&\{ \\
&\quad pow \leftarrow pow \times a \\
&\quad i \leftarrow i + 1 \\
&\} \\
\text{return } pow
\end{align*}
\]

Extra credit
5. [20 points] Consider the following algorithm.

\[
\text{ALGORITHM Mystery}(n) \\
//Input: A nonnegative integer n \\
S \leftarrow 0 \\
\text{for } i \leftarrow 1 \text{ to } n \text{ do} \\
&\quad S \leftarrow S + i \times i \\
\text{return } S
\]

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

b) [15 points] Compute the running time of this algorithm.