1. Prove the following polynomial is $\Theta(n^3)$.

\[ P(n) = 5n^3 + 10n^2 - 20n - 50 \]

(a) Prove $O(n^3)$:

\[
P(n) \leq 5n^3 + 10n^2, \quad n \geq 0
\]

\[
\leq 5n^3 + 10n^2 \left( \frac{m}{100} \right), \quad n \geq 100
\]

\[
\leq 5n^3 + 0.1n^3, \quad n \geq 100
\]

\[
\leq 5.1n^3, \quad n \geq 100
\]

(b) Prove $\Omega(n^3)$:

\[
P(n) = 5n^3 + 10n^2 - 20n - 50
\]

\[
\geq 5n^3 + 20n - 50
\]

\[
\geq 5n^3 - 20n \left( \frac{n}{100} \right)^2 - 50 \left( \frac{n}{10} \right)^3
\]

\[
\geq 5n^3 - 0.2n^3 - 0.05n^3
\]

\[
\geq 4.75n^3, \quad n \geq 10
\]
2. Consider the relation $R = \{(1, 1), (1, 3), (2, 2), (3, 1), (3, 2), (3, 3)\}$.

(a) Does the relation satisfy each of the following properties? Explain.

(12 points)

- Reflexive?
  
  YES, $(i, i) \in R$, $\forall i$.

- Symmetric?
  
  NO, $(3, 2) \in R$ but $(2, 3) \notin R$.

- Antisymmetric?
  
  NO, $(1, 3) \in R$ and $(3, 1) \in R$.

- Transitive? (Decide this by directly examining the ordered pairs. Do NOT use matrix multiplication for this part.)
  
  NO, $(1, 3) \in R$ and $(3, 2) \in R$, but $(1, 2) \notin R$.

- Partial Order?
  
  NO, not antisym and not trans.

- Equivalence Relation?
  
  NO, not symmetric and not trans.

(b) Show the matrix of this relation. Use matrix multiplication to decide if the relation is transitive. Explain.

(8 points)

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$


3. The following program computes \( \log_2 n \), assuming \( n \) is an integer power of 2.

```c
int LOG (int n) {
    int m, k;
    m = n;  k = 0;
    while (m > 1)
        { m = m/2;  k = k + 1  }
    return (k)
}
```

(a) Write a recursive version of this algorithm.

```c
int LOG (int n) {
    if (n == 1) return 0;
    T = LOG (n/2);
    return (1 + T);
}
```

(b) Let \( f(n) \) be the number of division operations performed by your recursive algorithm to compute \( \log_2 n \). Write a recurrence equation for \( f(n) \). Find the solution by repeated substitution.

\[
f(n) = \begin{cases} 
  0, & n = 1 \\
  1 + f\left(\frac{n}{2}\right), & n \geq 2
\end{cases}
\]

\[
f(n) = 1 + f\left(\frac{n}{2}\right)
\]

\[
= 1 + 1 + f\left(\frac{n}{4}\right)
\]

\[
= 1 + 1 + 1 + f\left(\frac{n}{8}\right)
\]

\[
\vdots
\]

\[
f(n) = K + f\left(\frac{n}{2^k}\right), \quad \text{where } n = 2^k, \quad K = \log n
\]

\[
f(n) = \log n
\]
4. Consider the following recurrence equation, where \( n \) is a power of 2.

\[
T(n) = \begin{cases} 
4T(n/2) + n, & n \geq 2 \\
0, & n = 1.
\end{cases}
\]

Prove by induction the solution is \( T(n) = An^2 + Bn \), and find the constants \( A, B \).

**Claim:** \( T(n) = An^2 + Bn \)

**Base, \( n = 1 \):** \( T(1) = 0 = A + B \)

To prove the claim for any \( n \geq 2 \), suppose the claim is true for \( n/2 \):

\[
T\left(\frac{n}{2}\right) = A\left(\frac{n}{2}\right)^2 + B\left(\frac{n}{2}\right)
\]

Then,

\[
T(n) = 4T\left(\frac{n}{2}\right) + n
= 4\left[A\left(\frac{n^2}{4}\right) + B\left(\frac{n}{2}\right)\right] + n
= An^2 + (2B + 1)n
\]

To make the latter equality, need \( 2B + 1 = B \)

So, we need:

\[
\begin{align*}
A + B &= 0 \\
2B + 1 &= B
\end{align*}
\]

This gives \( B = -1, \ A = +1 \).

So,

\[
T(n) = n^2 - n
\]
5. Consider the following divide-and-conquer recursive algorithm. Parameter $i$ is the starting index of the array, and $j$ is the ending index. The initial call for an array of size $n$ is FIND($A, 0, n - 1$).

Boolean FIND(int $A[]$, int $i$, int $j$) {
if ($i == j$) return FALSE;
$m = [(i + j)/2]$;
$C1 = $FIND($A, i, m$); \text{ // Left half}
$C2 = $FIND($A, m + 1, j$); \text{ // Right half}
$C3 = (A[i] \neq A[j])$
return ($C1 \lor C2 \lor C3$)
}

(a) Figure out what the algorithm does. When does the algorithm return TRUE? Be precise. Explain briefly how the algorithm works.

This algorithm returns true iff there exists any pair $i, j$ where $A[i] \neq A[j]$.
If $n = 1$ it returns false;
For $n \geq 2$, it makes two recursive calls and returns true iff
(1) There exist a pair in left half that are not eq.
or (2) \hspace{1cm} \text{right} \hspace{1cm}
or (3) An element from first half $\neq$ An element in 2nd half.

(b) Let $f(n)$ be the number of key comparisons performed by this algorithm for an array of size $n$. Assuming $n$ is a power of 2, write a recurrence equation for $f(n)$. Guess the solution of the recurrence and prove the correctness of it by induction.

$$f(n) = \begin{cases} 
0, & n = 1 \\
2 f\left(\frac{n}{2}\right) + 1, & n \geq 2 
\end{cases}$$

Claim: \[ f(n) = n - 1 \]

Base, $f(1) = 0$, $f(1) = 1 - 1 = 0$, base case is correct.
To prove for any $n \geq 2$, suppose true for $\frac{n}{2}$:
\[ f\left(\frac{n}{2}\right) = \frac{n}{2} - 1 \]
Then,
\[ f(n) = 2 f\left(\frac{n}{2}\right) + 1 = 2 \left(\frac{n}{2} - 1\right) + 1 = n - 2 + 1 = n - 1. \]