1. Prove that \( \sum_{i=1}^{n} \log i = \Theta(n \log n) \). 

(a) Prove \( O() \):

\[
\sum_{i=1}^{n} \log i \leq \sum_{i=1}^{n} \log n \leq n \log n
\]

(b) Prove \( \Omega() \):

\[
\sum_{i=1}^{n} \log i \geq \sum_{i=\lceil \frac{n}{2} \rceil}^{n} \log i \geq \sum_{i=\lceil \frac{n}{2} \rceil}^{n} \log \left[ \frac{n}{2} \right]
\]

\[
\geq \left\lceil \frac{n}{2} \right\rceil \log \left[ \frac{n}{2} \right] \geq \frac{n}{2} \log \left( \frac{n}{2} \right)
\]

\[
\geq \frac{n}{2} \left( \log n - 1 \right)
\]

\[
\geq \frac{n}{4} \log n,
\]

\( n \geq 4 \), \( \log n \geq 2 \)
2. 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;
    return (1 + log \( \frac{n}{2} \));
}
```

(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.

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

\[
f(n) = 1 + f\left(\frac{n}{2}\right)
= 1 + 1 + f\left(\frac{n}{4}\right)
= 3 + f\left(\frac{n}{8}\right)
\]

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

\[
f(n) = \log n
\]
3. 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 \( \text{FIND}(A, 0, n - 1) \).

```java
boolean FIND(int A[], int i, int j) {
  if (i == j) return TRUE;
  m = (i + j) / 2;
  C1 = FIND(A, i, m); \text{ // First half}
  C2 = FIND(A, m + 1, j); \text{ // Second half}
  C3 = (A[m] \leq A[m + 1])
  return (C1 \land C2 \land C3)
}
```

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

The algorithm returns TRUE iff \( A[i] \leq A[i+1] \), i.e.,

That is, if the array is sorted.

This is true iff

1. All elements in first half sorted,
   AND 2. All elements in second half sorted,
   AND 3. Last element in 1st half \( \leq \) first element in 2nd half.

That is, \( A[m] \leq A[m+1] \).

(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.

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

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

Base, \( n = 1 \): \( f(1) = 0 \)  \( f(1) = 1-1 = 0 \)

To prove for \( n \geq 2 \), suppose true for \( \frac{n}{2} \). Then,

\[
f(n) = 2 \cdot f\left(\frac{n}{2}\right) + 1
= 2\left(\frac{n}{2} - 1\right) + 1
= \frac{n}{2} - 1 + 1
= \frac{n}{2} + 1
= \frac{n}{2} + \frac{n}{2}
= n - 1, \text{ as desired.}
\]
4. Consider the following recurrence equation, where \( n \) is a power of 2.

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

Prove by induction that the solution is as follows, and find the constant \( A \).

\[ T(n) = An \log n. \]

**Base, \( n = 1 \):** \( T(1) = 0 = A - 1 \cdot \log 1 \) \( \checkmark \)

To prove for any \( n \geq 2 \), suppose the case \( \frac{n}{2} \):

\[ T\left(\frac{n}{2}\right) = A \frac{n}{2} \cdot \log \frac{n}{2} \]

Then,

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

\[
= 2 \left[ A \frac{n}{2} \left( \log \frac{n}{2} \right) \right] + n
\]

\[
= 2 A \frac{n}{2} \left( \log n - 1 \right) + n
\]

\[
= An \log n + (-A+1) n
\]

Need \[ An \log n + (-A+1) n \]

Need \[ A \log n \]

So, we proceed:

\[ T(n) = An \log n \quad \text{an} \quad A = 1 \]

\[ \therefore T(n) = n \log n \]
5. Consider the relation \( R = \{(1, 1), (1, 2), (2, 1), (3, 3)\} \).

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

- Reflexive?

  No, \((2, 2) \notin R\)

- Symmetric?

  Yes, \( \forall i, j, (i, j) \in R \) then \((j, i) \in R\).

- Antisymmetric?

  No, since \((1, 2) \in R\) and \((2, 1) \in R\).

- Transitive? For this part, you must decide it by directly examining the ordered pairs, and not using matrix multiplication.

  No, because \((2, 1) \in R\) and \((1, 2) \in R\)

- Partial Order?

  Not reflex and not transitive.

- Equivalence Relation?

  Not reflex and not antisymmetric and not transitive.

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

\[
A = \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

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

Not transitive because

\[
\]