Answer all 5 problems, each worth 20 points.

Be concise in describing your algorithms. If you use a known algorithm, then use it as a black-box (subroutine) without going into details of how that known algorithm works. Avoid useless rambling about known algorithms, as that is viewed negatively!

1. Consider multiplication of two matrices \( A \) and \( B \), each of size \( n \times n \). Let the product be matrix \( C = A \times B \). Suppose \( n = 2^k \) for some integer \( k \). Consider the classical matrix multiplication algorithm implemented using a divide-and-conquer approach. Each matrix is divided into 4 quadrants, each of size \( n/2 \times n/2 \). Then matrix multiplication is carried out as if all quadrants were single elements.

\[
\begin{bmatrix}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{bmatrix}
= \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\times
\begin{bmatrix}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{bmatrix}.
\]

(a) Write down the expression for each of the quadrants in the product matrix \( C \). (The expression for \( C_{11} \) is already provided.)

\[C_{11} = A_{11} \times B_{11} + A_{12} \times B_{21}\]

(b) Let \( T(n) \) be the total number of simple arithmetic operations (i.e., number of multiplications and additions) for this divide-and-conquer algorithm to multiply two \( n \times n \) matrices. Write a recurrence equation for \( T(n) \). Provide a brief explanation for each term.

(c) Use Master-Theorem to obtain the solution.

2. Given two nodes \( a \) and \( b \) in a binary tree. Describe an algorithm (psuedocode) to find the Nearest Common Ancestor (NCA) of the two nodes. Analyze the time complexity in terms of the number of nodes, \( n \), and the height of the tree, \( h \).

Assume each node in the tree has three pinters: Parent, Leftchild, and Rightchild. In addition, assume each node contains an attribute that gives its Depth (i.e., distance from the root).

3. Given a directed graph, represented by its Boolean adjacency matrix \( A \), where

\[A[i,j] = \begin{cases} 
1, & \text{if } (i,j) \text{ is an edge, or } i = j, \\
0, & \text{otherwise}.
\end{cases}\]

(a) Suppose we compute the Boolean square of the matrix, \( A^2 = A \times A \), where the the product matrix is also Boolean. That is,

\[A^2[i,j] = \bigvee_{k=1}^{n} (A[i,k] \land A[k,j]).\]
In simple words, explain what each entry $A[i, j]^2$ gives in terms of the graph. Provide your reasoning.

(b) Let $n$ be the number of vertices in the graph. Explain what the matrix $A^n$ will give in terms of the graph. Give a simple inductive reasoning.

(c) Outline an efficient method for computing $A^n$. For simplicity, assume $n$ is a power of 2. Analyze the time complexity.

4. Given an array $A[0..n-1]$ of $n$ random real-values. We want to determine if all values in the array are distinct.

   (a) Describe an algorithm with a good worst-case time performance. What is the worst-case time complexity?

   (b) Outline an algorithm with a good average-case performance. What is the average and worst-case time complexity of this algorithm?

   (c) Now suppose the array is integer values. What is the best algorithm for this case? What is the average and worst-case time complexity?

5. Given an array $A[0..n-1]$ of $n$ random real-values, and an integer $t$. We want to find the $2^k$th smallest element, for $k = 0, 1, 2, \ldots, t$. (For example, for $t = 3$, the algorithm finds the smallest, second smallest, 4th smallest, and 8th smallest.) Describe an algorithm for this problem with the worst-case time $O(n)$. Analyze the time complexity.