1. (20 points) For each of the parts below, give a short answer (a few words or symbols, or perhaps a short sentence).

(a) Use quantifiers to express the proposition that, for all positive real numbers, there exists another positive real number which is smaller. Recall that the set of positive real numbers is given by the symbol \( \mathbb{R}^+ \).

(b) Give quick definitions of the following three types of trees, focusing on what makes them different from each other: binary tree, binary search tree, B tree.

(c) Why does an inductive proof need a base case? That is, why don’t we simply prove the inductive step and that is enough?

(d) Explain the difference between \( O(n^2) \) and \( \Theta(n^2) \).
2. (10 points) A heap is optimized to find either the minimum or maximum in a data set. For instance, in a max-heap, you can find the maximum in $O(1)$ time. (Removing it, while still cheap, is a little more expensive.)

How would you find the **minimum** value in a max-heap? What is the asymptotic time cost of doing this search? (Don’t worry about the cost of removing the node once you find it.)

3. (10 points) Imagine that you have two asymptotic statements:

\[ f(n) = X(g(n)) \]
\[ f(n) = Y(g(n)) \]

where $X, Y$ are asymptotic operators (big-O, Theta, little-o, big-Omega, and little-omega).

Suppose that, based on those two, you then deduce a third expression:

\[ f(n) = Z(g(n)) \]

If $X, Y, Z$ are all different operators, then identify what each operator must be. That is, what were the two operators which you started with, and what was the third that you deduced? Explain your answer.
4. (10 points) Use the Master Method to solve the following recurrences. For each one, show the values of $a$ and $b$, identify $f(n)$, identify which of the three cases it is, and give the solution. If the recurrence cannot be solved, explain why.

(a) $T(n) = 16T\left(\frac{n}{4}\right) + n^2$

(b) $T(n) = 2T\left(\frac{n}{2}\right) + n \log n$
5. (20 points) (a) Describe how Quicksort works. (Just give the overview, don’t worry about explaining its worst-case behavior.)

(b) Describe how Merge Sort works. Describe it as an iterative algorithm (that is, using loops inside a single method) instead of a recursive one.
6. (10 points) Professor Badeye Dias has found a way to simplify Merge Sort: instead of using Divide and Conquer (and thus merging blocks together in a sort of tree), his algorithm immediately merges each new block into a large, growing list of sorted data. For instance, if the block size is 16 elements, then he first makes two blocks of size 16, and then merges them into a single block of size 32; he then makes another block of size 16, and immediately merges it in, to make a block of size 48.

This algorithm does $O(n)$ many merges - where as ordinary Merge Sort does a lot more. However, the Professor’s idea doesn’t work well at all.

Argue that the Professor’s new version of Merge Sort takes $O(n^2)$ time to run.
public class Node
{
    public int key;
    Node left, right;

    public Node rotateRight()
    {
        // Implementation goes here
    }
}
8. (10 points) Describe at least two different strategies for handling collisions in a hash table. Give an overview of how each works, and the tradeoffs involved between them.

9. (10 points) Explain how the garbage collector in a garbage-collected language (such as Java) might use graph algorithms to determine which objects are garbage. Would it likely use a graph or a digraph, and why? Compare and contrast the heap and the stack; how would they be represented in this graph or digraph?
10. (15 points) We have said (without proof) that, in a binary heap, node $k$’s children - if it has any - are the nodes $2k + 1$ and $2k + 2$.

Use structural induction to prove this assertion.

**Hint:** I can think of an easy proof using the “add a node or two” technique. You don’t have to use it, but I encourage you to.
11. (10 points) In Breadth First Search, each of the vertices is always in one of three states. Describe what the states are and what they mean. Be sure to clear how and when a vertex transitions from one state to another.

12. (5 points) What is the asymptotic time cost of inserting a new value into a max-heap? Describe the algorithm used to accomplish this.

13. (10 points) Why is it impossible to build a non-deterministic computer in practice?