Question 1:
How many nodes can there be in:

a) A left-complete binary tree of height $h$?

b) A perfect binary tree of height $h$? (In a perfect tree, all nodes have 0 or 2 children, and all leaves are on the same level.)

c) A perfect ternary tree of height $h$? (In a ternary tree, each node may have up to 3 children.)

Question 2:
Many recursive, and even non-recursive, algorithms can be modeled by a recurrence relation that has the following form:

$$f(n) = a*f(n/b) + c \text{ for all } n > 1, \text{ where } a, b, \text{ and } c \text{ are positive constants}$$

a) Give the closed form (solution) of this recurrence, simplifying it as far as possible.

b) Why do we not need to specify $f(1)$ to know the growth rate of the recurrence?

c) What happens if $a=1$?
Question 3:
a) Describe in 2-3 sentences how mergesort works on an array-based list of numbers.

b) Give a recurrence for the number of comparisons mergesort performs in terms of the number of items in the list. Indicate the purpose of each part of the recurrence.

Question 4:
a) A worst-case AVL tree, also known as a Fibonacci tree, is an AVL tree in which only the leaves have a balance factor of 0. Give a recurrence for the number of Fibonacci trees having height h. (You do not need to solve the recurrence.)

b) Explain why performing an AVL single rotation on a binary search tree does not change the order of nodes visited by an inorder traversal of the tree.
Question 5:

a) What property do nodes in a minimum heap have, compared to a vanilla binary tree?

b) Consider the following heap:

![Diagram of a tree heap]

Show how this tree heap (image) could be represented in array form.

c) Suppose you have an array heap as in part B, and that k is the index of a node in the heap. Give formulas for the index of the left and right children of node k.
d) Write a pseudocode algorithm to remove the smallest item from an array heap.

Question 6:

a) Consider the following binomial heap (i.e. a binomial forest that has the heap property):

Show the result of inserting the value 2 into this heap, and all intermediate merges.
b) Explain why finding the minimum value in a binomial forest is $O(\log n)$, where $n$ is the number of nodes in the heap.

Question 7:

a) Normally, iterating over a hash map returns the elements in an unpredictable order. John Q. Programmer needs to iterate over the elements in the order he inserted them. Explain how to modify a generic hash map to grant this capability. The modified structure MUST have constant lookup time; you may sacrifice performance elsewhere if necessary, but do so as little as possible.

b) Give the growth rates for insertion and deletion on the new structure in terms of the number of nodes.