1. It is not uncommon to combine two sorting algorithms to fulfil the performance requirements of the problem at hand. Discuss the scenario a) where b) Why and c) how, we will use the following combinations.
   a) Insertion Sort and Merge Sort 1.5
   b) Radix Sort and Counting Sort 1.5
   c) Quick Sort, Merge Sort and External Sort 1.5

2. Consider a situation, where we have a list of N numbers to be handled. To get to the solution we use the divide and conquer strategy. Every time the list is divided into one-third and two-third. The amount of work required to break the list into these two sets is N. The process goes on until we recursively reach at one element each in the list. Work required to combine the list to reach the final solution is O(1). Write a recurrence for this situation. Solve the recurrence using the recurrence tree and find out the time complexity. 1.5

3. We are given a binary search tree. We want to find out the i^{th} element in the inorder sequence of this tree.
   But, for this we don't want to use the naïve algorithm which finds inorder sequence and then prints the i^{th} element in the sequence.
   So, write an alternative algorithm (with proper comments) that gives us the i^{th} element in the inorder sequence of the BST and does not takes more time than the naïve algorithm. 2

4. Use a Programming Language of your choice to write a program or a Pseudo code to write an algorithm for Insertion and deletion of an element in the circular priority queue. (Define your notion of Priority for your answer).
   2