1. Provide solutions (using big-Oh or big-Theta) for the following recurrence relations.

   (a) \( T(n) = 9T(n/3) + n^2 \)
   (b) \( T(n) = 5T(n/3) + n \)
   (c) \( T(n) = 5T(n/3) + n^3 \)

   [9 points]

2. Into an initially empty AVL tree, insert the following values: 22, 30, 25, 10, 8, 6, 12, 15, 24, 20, 18.

   [11 points]

3. Insert the values above into an initially empty 2-3-4 tree.

   [11 points]

4. Consider the following routine which takes an array of integers \( A[1], A[2], \ldots, A[n] \) and constructs a BST containing them.

   ```python
   buildBST(array A)
   BST T
   for i=1 to n
       T.insert(A[i])
   return T
   ```

   (a) What is the worst case time of this routine? Explain.
   (b) What is the average case time of this routine? Explain.
   (c) What is the best case time of this routine? Explain.

   [9 points]

5. For this question we allow a BST to contain duplicate values: for example, there may be three nodes \( p \) for which \( p.value=5 \). If node \( p \) stores a 5, another 5 may be in the left subtree of \( p \), but not in the right subtree.

   Write a short routine `countNodes(node p, int k)` which determines the number of nodes in the subtree rooted at \( p \) containing the value \( k \).

   [10 points]

Total: 50 points