Tutorial Session 9

1. Construct an expression tree for the expression (a + b * c) +((d * e + 1) * g). Give the outputs when you apply preorder, inorder, and postorder traversals.
2. Create a binary search tree for the following numbers starting from an empty binary search tree. 45,26,10,60,70,30,40

   Delete keys 10,60 and 45 one after the other and show the trees at each stage.
3. For the following pairs of traversals, draw the tree that is consistent with the traversal tree. Also, give the other traversals.

   Pair 1

   In order: C O M B N A H K S D V T J

   Preorder: K B O C M A N H T D S V J

   Pair 2

   In order: D B F E A G C L J H K

   Postorder: D F E B G L J K H C A
4. Show a binary search tree (BST) which is a complete binary tree and have a total of 11 nodes. Data value stored in each node is an integer. If this cannot be done (i.e., the requested tree does not exist), explain why.
5. Discuss the average-case and worst-case time complexity of various BST operations (e.g., search, insertion, deletion) based on the tree's height.
6. Describe one or more real-world scenarios or applications where binary search trees are commonly used. Explain why BSTs are suitable for these applications.
7. Discuss potential challenges or limitations when using BSTs in real-world systems.