Unique Paper Code	**: 42347902**
Name of Course	**: B.Sc Prog. Computer Science/ B.Sc Mathematical Science**
Name of the Paper	**: Analysis of Algorithms and Data structures**
Semester	**: V**
Year of Admission	**: Upto 2018**

Duration: 3 Hours **Maximum Marks: 75**

Instructions for Candidates:

1. Answer any **FOUR** questions.
2. All questions carry equal marks.

1. Which data structure would you chose for the following tasks and Why? Justify your answer in each of the following case: Case 1: to implement undo/redo operation in a text editor; Case 2: bookmarks tab of a web-browser; Case 3: you need to get out of a maze (without using recursion), for which you are required to store the path that you are currently exploring and be able to go back one step in case of a dead-end and then explore a new possibility from there; Case 4: to store two sorted lists of strings and support their merging into one, in-place (i.e., without creating a copy of the lists); Case 5: write a software for a call center, in which whenever a client calls, his call record should be stored if there is no free operator to pick it up; Case 6: to store the stations of a public transportation line in which new stations can be added at both ends of the line but not between existing stations. Transportation line can be traversed in both the directions; Case 7: create table of contents, i.e., index page of a book. It must show every chapter, its sections and corresponding sub-sections, to all levels.

2. Given a binary search tree and the following pseudocode to traverse the tree. Determine in what order is the function traversing the tree? Modify it to count the number of nodes in the tree.

```
fun traverse(Node t)
        if t==NULL
                return
        else
                traverse(right child of t)
                print(value of t)
                traverse(left child of t)
```

Also, determine which of the following statements is/are false about tree traversals? Justify your answer.

a. If values 1,2,3,4,5,6 are inserted in the given order in a Binary search tree, then in-order, pre-order and level-by-level traversal are all same.
b. If values 6,5,4,3,2,1 are inserted in the given order in a Binary search tree, then in-order and pre-order traversal are same.
c. If values 6,5,4,3,2,1 are inserted in the given order in a Binary search tree, then pre-order and level-by-level traversal are same.

d. If values 6,5,4,3,2,1 are inserted in the given order in a Binary search tree, then in-order and post-order traversal are same.

3. We want to sort an array of 1000 integers, in which every element in its correct place except 10 elements. Arrange the following sorting techniques in decreasing order of preference for sorting: Insertion, Quick, Merge, Count. Justify your preference with suitable arguments. Suppose the condition now reverses, i.e., only 10 elements are in their correct place. Specify the order of preference again with justifications.

4. Consider the following modification in (binary) search algorithm on an array A[1...n], sorted in increasing order. The array is divided into three equal parts. Instead of finding one middle index, we find two middle indices, mid_1 and mid_2. If element at mid_1 or mid_2 equals the element 'x' that we are trying to search, we stop. Else, we do either of the following: search x from A[1...mid_1 -1] if x is less than element at mid_1 or search x from A[mid_1 +1...mid_2 -1] if x is greater than element at mid_1 but less than element at mid_2 or search x from A[mid_2 +1 ... n] if element at mid_2 is less than x. Write an iterative algorithm for the above mentioned search strategy. What should be the running time of the algorithm? Also, write the recursive version of the algorithm.

5. What value should be chosen as pivot in the following array so that when we apply quick sort to it, the first call to partition should belong to worst-case scenario?
-5,-3,1,4,2,-6,1,5
Justify your answer. Also, show steps in partitioning the array around the chosen pivot.
Can you use Count sort to sort the above array? If yes, modify the algorithm to do so. If no, justify your answer by giving suitable arguments.

6. Write algorithms to implement the following operations on a circular doubly linked list:
   i. Search for an element x in the linked list.
   ii. Using the function implemented in part (i) above, insert an element x after an element y in the linked list.
   iii. Using the function implemented in part (i) above, replace an element x with an element y in the linked list.
   iv. Remove i^{th} element from the linked list.

   Using the algorithms above, perform the following functions. Also, specify which algorithms (out of above mentioned) along with their parameters, would you use for performing the tasks mentioned below.
   Insert every character of the word "ALGORITHMS AND DATA STRUCTURES" as a node in an initially empty list. Now, scan the list and replace ' ' (space) character with a '#'. Change the case of value of every alternate node, i.e., AlGoRiThMs and so on. Delete i^{th} node of the list where i = (length of list)%10 from back. Print the final list.