Data Structure Laboratory Assignment-4

B.E. (Information Technology), 1st Year, 2nd Semester

1. Write a programme using C functions that will perform the following operations on an ***array representation*** of a binary tree (an estimation of the number of nodes in the tree will an input from the user):
    a. Inserting a data in the tree to make the tree as complete as possible.
    b. Searching a particular data in the tree, or return null in case the data is not found.
    c. Removing the first occurrence of a data in the tree, or return null if the data is not present.
    d. (optional) Finding height of the binary tree. No Math function is to be used.
    e. Verify the relation between number of nodes and height of a tree.
    f. Pre-order traversal of the tree.
    g. In-order traversal of the tree.
    h. Post-order traversal of the tree.
2. Define a C-structure that represents a node in a binary tree. Using this structure, create a binary tree and perform the operations as specified in question 1.
3. Using the C-structure defined in question 2, create a binary search tree that will perform the same operations as in question 1.
4. Define a C-structure that corresponds to a node in an AVL tree. Using this structure, write a programme with functions, that will perform the operations in the AVL tree as in question 1.
5. (Linear Searching) Write a C-programme that will perform linear search in a dynamic array of data.
6. Write a C-programme that will perform the following sorting methods on a dynamic array of data:
    a. Bubble sort.
    b. Insertion sort.
    c. Selection sort.
7. Write a C-programme that will perform binary search on a dynamic array of data.
8. (Optional) Binary search method can be improved for repeated searching by storing the position of the data that has been retrieved in the last successful search. Write a programme to implement the modified Binary search method on a dynamic array of data.
9. Write a programme that will perform Merge sort on a dynamic array of data. The size of the array will be one of the inputs from the user.
10. Let L be a list of values which is to be stored in an array named *Cache*. Let m be size of Cache. The values from L is to be stored in Cache, where the index in which a value to be stored will be determined using hashing. Let the hash function to be used be: h(v)= v mod m, where v is the value to be inserted in Cache. Assume ***m=100***. Perform the following operations (you may use the file I/O functions) by writing corresponding C programme(s):
    a. Given 1000 values (in the file named NumberGen_4_digit.txt) use the hash function, h, to find the index in Cache where a value is to be stored. Collision resolution is to be solved using linear probing. Note that if linear probing leads you to (m-1)-th index, i.e. 99 in this question, and there is already an inserted value, then search at 0th index, 1st index, and so on. Use a variable to count number of the collision, hence determine the collision rate for the given data-set.
    b. Repeat the hashing procedure of part-a, if chaining is used to accommodate the values having same hash value. Redefine Cache accordingly. Also find the average length of chaining for the specific data-set.

c. Repeat the hashing procedure of part-a, if the hash function is described as: *“Select first two digits of ‘v’, let the value of the two digit be x. If 0≤v≤10, then x=v. Then x will give the index where v is to be inserted.”* Determine the collision rate for this scheme and the given data-set.