**Dept. of ECE**

**Fall 2018**

**Assignment-2**

**Course: Data Structure and Algorithm (CSE225)**

**Submission Date: 11/12/2018**

1. It is often handy to store a binary tree in a file. Assume each node in the binary tree contains a character string. Assume also that all operations you need on strings are provided. For example, you do not need to design algorithms to test if a string equals ".", to write a string into a file, or to read a string from a file. To create the file in preorder file format do a preorder traversal of the tree, when a node is visited put the character string in the file followed by a newline and when a null is visited put a dot, ".", followed by a newline. For example, the tree

                              a

                            /   \

                           b     c

                          /    /   \

                         d    e     f

is stored as the file "a b d . . . c e . . f . ." where spaces indicate newlines.

* 1. Design an algorithm which outputs the preorder file format of a binary tree given a pointer to the root of a binary tree. Assume the binary tree has nodes with fields "data", "left\_child", and "right\_child".
  2. Design an algorithm which takes a preorder file format of a binary tree and produces the binary tree. Hint: an effective approach is to design a recursive function that processes a sequence of lines in a file and returns a binary tree.

1. Using an in-order traversal it is quite easy to output all the data in a binary search tree in order. Design an algorithm which when given a binary search tree and two numbers x and y outputs all the data items z in order with the property that x <= z <= y. If the tree has height d then the total time to output should be O(d + k) where k is the number of data items output by the call. You should argue that your algorithm takes that much time. Hint on the analysis:  the visited nodes are found to the right of one path in the tree and to the left of another.