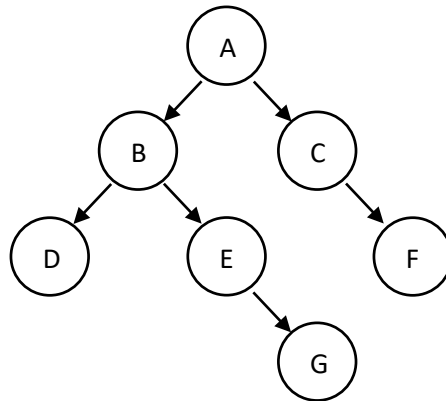


American University of Armenia
CS121, Data Structures
PSS 9

Trees

1. Given the following tree, describe the order in which the nodes are visited during:
 - Inorder traversal
 - Preorder traversal
 - Postorder traversal
 - Breadth First Search



2. Two ordered trees T1 and T2 are said to be isomorphic if:
 - Both T1 and T2 are empty
 - Both T1 and T2 consist of a single node
 - The roots of T1 and T2 have the same number of subtrees, and i-th such subtree of T1 is isomorphic to the i-th subtree of T2

Give a method that tests whether two given trees (that are assumed to be ordered) are isomorphic. What is the runtime of the algorithm?

Binary Trees

1. Roman position is described as position p in binary tree T, such that the difference between the number of descendants in p's left and right subtrees is at most 5. Give a method for finding Roman position in T whose time complexity is $O(n^2)$.
2. Mirror of a Binary Tree T is another Binary Tree M(T) with left and right children of all non-leaf nodes interchanged. Add a method mirror() in AbstractBinaryTree that will mirror the entire tree.
3. Add a method containsSubtree() in AbstractBinaryTree that given a binary tree T, checks whether T is a subtree of the one defined in this class.

LinkedBinaryTree

1. Add a method pruneSubtree(p) in LinkedBinaryTree that removes the entire subtree rooted at position p and maintains an accurate count of the size of the tree. What is the runtime of this method?
2. Add a swap(p, q) method in LinkedBinaryTree that has the effect of restructuring the tree so that the node referenced by p takes the place of the node referenced by q, and vice versa (the problem cannot be solved by simply swapping the elements - Positions p and q must be swapped; moreover, the subtrees of p and q should not be affected by the swap).

If time permits

1. Draw the unique binary tree that has the following preorder and inorder traversals:
Preorder: A, B, D, E, C, F, G, H
Inorder: E, D, B, A, G, F, H, C
2. Given a Binary Tree T, find the difference between the number of nodes at odd levels and the number of nodes at even levels.
3. Give a method for finding Roman position in tree T in $O(n)$ time.