

SET 5

Binary tree

Binary tree as ADT

1. ADT Binary tree

operations on binary tree

(recursive / non-recursive using stacks and/or queues)

1. Create a binary tree on levels (specification & pseudocode)
2. Create a binary tree in pre-order (specification & pseudocode)
3. Given an element, **search** it in a binary tree (return its position). Write a recursive and then a non-recursive function (specification & pseudocode)
4. Given a node (by its position) in a binary tree, find its **height**. Write a recursive and then a non-recursive function. (specification & pseudocode)
5. Given a node (by its position) in a binary tree, find its **parent**. Write a recursive and then a non-recursive function. (specification & pseudocode)
6. Given a binary tree, write a function to find the **depth** of a given node. Write a recursive and then a non-recursive function. (specification & pseudocode)
7. Write a function SwapTree that takes a binary tree and swaps the left and the right children of every node.
8. Write a subalg. that takes as input a binary tree T, and outputs the tree that results after removing all leaves from T.
9. A binary tree is **strict** if any node have either both empty or both non-empty subtrees. Write a function to decide whether a binary tree is strict.
10. Verify if a binary tree is (Almost) complete