152113022 VERİ YAPILARI LABORATUVARI LAB

LAB WORK 9

10 December 2024

Objectives:

• Splay Tree, BST, Traversal

There is one operation called splaying, which makes Splay Tree different from AVL Tree. A splay tree contains all the operations of a binary search tree, like insertion, deletion, and searching. But it also contains one more operation, which is called splaying. In a splay tree, every operation is performed at the root of the tree. Splaying is the process of bringing an element to the root by performing suitable rotations. By splaying elements in the tree, we can bring more frequently used elements closer to the root of the tree so that any operations like insertion, searching, and deletion can be performed quickly.

Question 1. Fill in the functions given below in accordance with the Splay Tree structure. At the end of the process, create a tree and fill this. Then perform the BST operation for the Splay Tree. Use the preOrder, Inorder, postOrder methods to display the tree on the screen.

Question 1.a.
- Define Tree Node for Splay Tree.

Question 1.b.
- Define Right Rotation for Splay Tree.
  (A utility function for right-rotating a y-rooted subtree)

- Define Left Rotation for Splay Tree.
  (A utility function for right-rotating a x-rooted subtree.

Question 1.c.
- Define Splay function. The Splay function should contain the functions given in the table below.
  (This function modifies the tree and returns the modified root.)

Question 1.d.
- Define BST Search.

Question 1.e.
- Define preOrder, Inorder, postOrder
  (The utility functions to print preorder traversal of the tree.)

Hint!

Key lies in left subtree	Key lies in right subtree
Zig-Zig (Left Left) Zig-Zag (Left Right)	Zag-Zig (Right Left) Zag-Zag (Right Right)

Question 2. Find min/max value for subtree.

TreeNode* subtree_max(TreeNode *subRoot)
{
}
	TreeNode* subtree_min(TreeNode *subRoot)
	{
	}

Good Luck!