

WIA1002 Data Structure

Tutorial 7: Binary Search Tree

1. Draw the BST based on the input 50, 30, 25, 71, 80, 99, 40, 1, 7, 5. Then, list the items in the order of pre-order, in-order and post-order.
2. Create the BSTNode.java and BST.java file (only the methods required based on the lecture notes). Then, perform the following:
 - a. Insert the characters F, A, C, U, L, T, and Y into the BST.
 - b. Print the number of elements in the BST
 - c. Display the BST elements (PREORDER, INORDER, POSTORDER)
 - d. Remove character L from BST
 - e. Display the BST elements (PREORDER, INORDER, POSTORDER)

Example output:

```
The number of elements in the tree : 7
The tree elements - PREORDER : F --> A --> C --> U --> L --> T --> Y -->
The tree elements - INORDER : A --> C --> F --> L --> T --> U --> Y -->
The tree elements - POSTORDER : C --> A --> T --> L --> Y --> U --> F -->
Remove 1 Character L
The number of elements in the tree : 6
The tree elements - PREORDER : F --> A --> C --> U --> T --> Y -->
The tree elements - INORDER : A --> C --> F --> T --> U --> Y -->
The tree elements - POSTORDER : C --> A --> T --> Y --> U --> F -->
```

3. Create an ADT Binary Search Tree named **ArrayBST**. The ADT consists of the following method. Given the maximum size of the ADT is 20. Left child of node K is, $2K+1$; Right child of node K is, $2K+2$. If the node is root, $K=0$, left child is in position 1 and right child is in position 2.
 - a. Constructor
 - b. isEmpty
 - c. getSize
 - d. getHeight
 - e. addNode
 - f. inOrder
 - g. preorder
 - h. postOrder
 - i. removeNode

Example output:

```
BST Implementation Using Array
The number of elements in the tree : 7
The tree elements - PREORDER : F A C U L T Y
The tree elements - INORDER : A C F L T U Y
The tree elements - POSTORDER : C A T L Y U F
Remove 1 Character L
The number of elements in the tree : 6
The tree elements - PREORDER : F A C U T Y
The tree elements - INORDER : A C F T U Y
The tree elements - POSTORDER : C A T Y U F
```