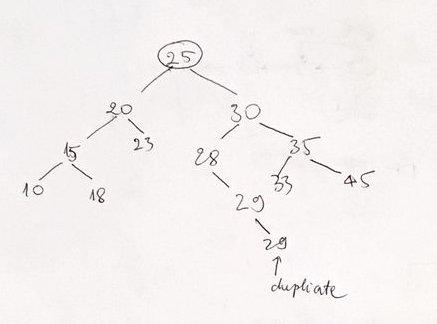
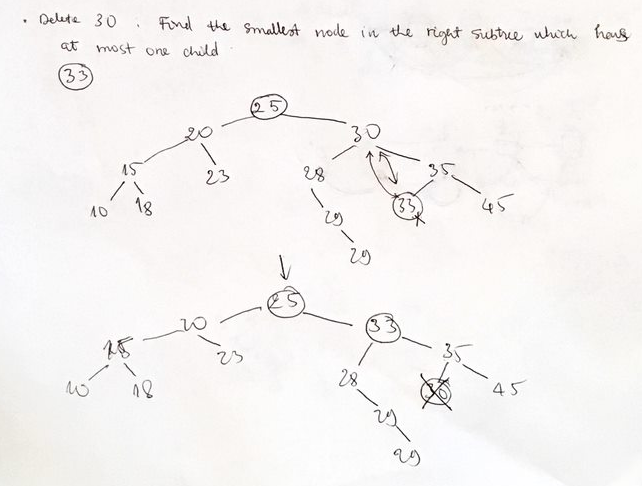
**Course: Algorithm**  
**Prof. Prem Nair**  
**Student: Binh Van Tran**  
**ID: 986648**  
**Homework: Lab 9**

1. *Build a BST by inserting one element at a time. After each insert, please draw a picture. For this assignment, you can submit hand drawn pictures*

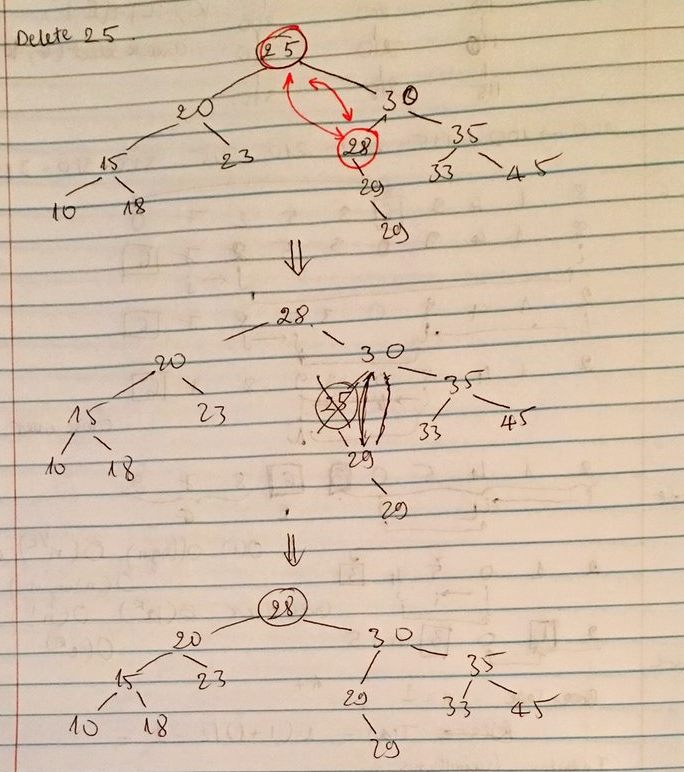
25, 20, 30, 28, 29, 15, 18, 23, 10, 35, 45, 33, 29



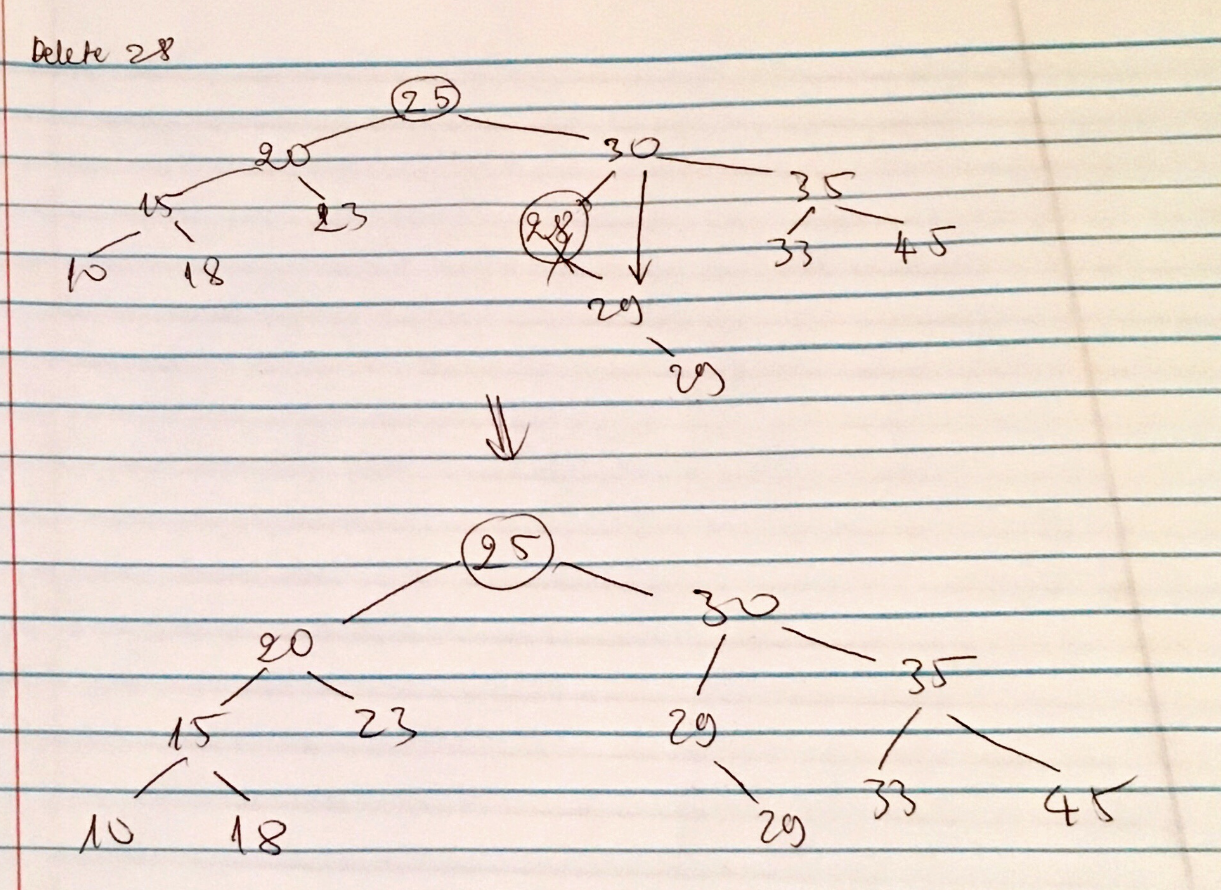
1. *Question 2 – Deletion*
2. Start from the BST you built. Delete 30



1. Start from the BST you built. Delete 25



1. Start from the BST you built. Delete 28



1. *Question 3 – Traversal* 
   1. Start from the BST you built. Preorder traversal

Root – Left – Right

25 20 15 10 18 23 30 28 29 29 35 33 45

* 1. Start from the BST you built. Post-order traversal

Left – Right – Root

10 18 15 23 20 29 29 28 33 45 35 30 25

* 1. Start from the BST you built. In-order traversal

Left – Root – Right

10 15 18 20 23 25 28 29 29 30 33 35 45

1. *Question 4 -* Write a recursive function to
   1. *Count the node of the BST*

***Algorithm*** *countNodes(T)*

***Input*** *BST T*

***Output*** *number of nodes in T*

**if** T == null **do**

**return** 0

**return** 1 + countNodes(T.left) + countNodes(T.right)

* 1. *Count the leaves of BST*

***Algorithm*** *countLeaves(T)*

***Input*** *BST T*

***Output*** *number of leaves in T*

**if** T == null **then**

**return** 0

**if** T.left == null && T.right == null **then**

**return** 1

**return** *countNodes*(T.left) + *countNodes*(T.right)

* 1. *Create a mirror image of BST*

***Algorithm*** *createMirrorImage(T)*

***Input*** *BST T*

***Output*** *mirror image of T*

**if** T != *null* **then**

tmp ← T.left

T.left ← T.right

T.right ← tmp

*createMirrorImage*(T.left)

*createMirrorImage*(T.right)

**return** T