

## Data Structure and Algorithm [Assignment 3]

**Submission to be done on Portal**  
**Deadline: Saturday, June 12, 2021, 11:50pm**

### Task no 1

To delete the given node from the Binary Search Tree (BST). We should follow the given rules.

**1. Leaf Node**

If the node is leaf (both right and left child will be NULL), remove the node directly and free its memory.

**2. Node with left child**

If the node has only left child (right will be NULL) make the parent node point to the left node.

**3. Node with right child**

If the node has only right child (left will be NULL) make the parent node point to the right node.

**4. Node has both left and right child's**

If the node has both left and right child,

- Find the largest node in the left subtree (in order predecessor). Say max
- Make node->data=max
- Again, delete the max node

Now your task is to write a C++ function for the above 4<sup>th</sup> case of deletion, in which node have both right and left child's. You have to submit the .Cpp file for the given problem.

**Note: your code must be non-recursive**

**Void delete4node(Node \* root)**

```
{  
  
}
```

## Task no 2

Write a C++ function which should be able to return the height of the Binary Search Tree (BST). You have to submit the .Cpp file for the given problem.

**Int BST\_height(Node \* root)**

```
{  
}
```