

# CS5800: Algorithms Spring 2018

## Assignment 5.3

Saptaparna Das

March 16, 2018

### Algorithm to check if two bsts are identical:

1. check root of the trees. If both are null then return true.
2. If one of the roots is null, return false.
3. If both the roots contains same value, recursively check left and right subtree of both the roots.  
Else return false

**Analysis:** The time needed to do check if both tree are identical, it taken  $O(n)$  time since each node is visited and compared once.  $n$  is number of nodes of smaller tree in case of two different sized trees. Space complexity is  $O(1)$ .

### Algorithm to create copy of binary search tree:

1. If root the tree is null, return null.
2. Create a new tree with root same as given binary search tree.
3. Recursively set left of new root same as left of given bst.
4. Recursively set right of new root same as right of given bst.

**Analysis:** The time needed to create a copy of bst is  $O(n)$  since every node is processed once to create new bst. Space complexity should constant or  $O(1)$