**CS2023 - Data Structures and Algorithms**

**Take Home Assignment**

Week – 06

Submission by: Sajeev Kugarajah (210554M)

1.

Delete(root, value):

if root = null then

return null

else if value < root.data then

root.left = Delete(root.left, value)

else if value > root.data then

root.right = Delete(root.right, value)

else

if root.left is null and root.right is null then

delete root

root = null

else if root.right is null then

temp = root

root = root.left

delete temp

else if root.left is null then

temp = root

root = root.right

delete temp

else

temp = findMax(root.left)

root.data = temp.data

root.left = Delete(root.left, temp)

end if

end if

return root

findMax(root):

if root.right is not null then

return findMax(root.right)

return root

2.

Expected time complexities for operations

insertion – O(log n)

search – O(log n)

deletion – O(log n)

in-order traversal – O(n)

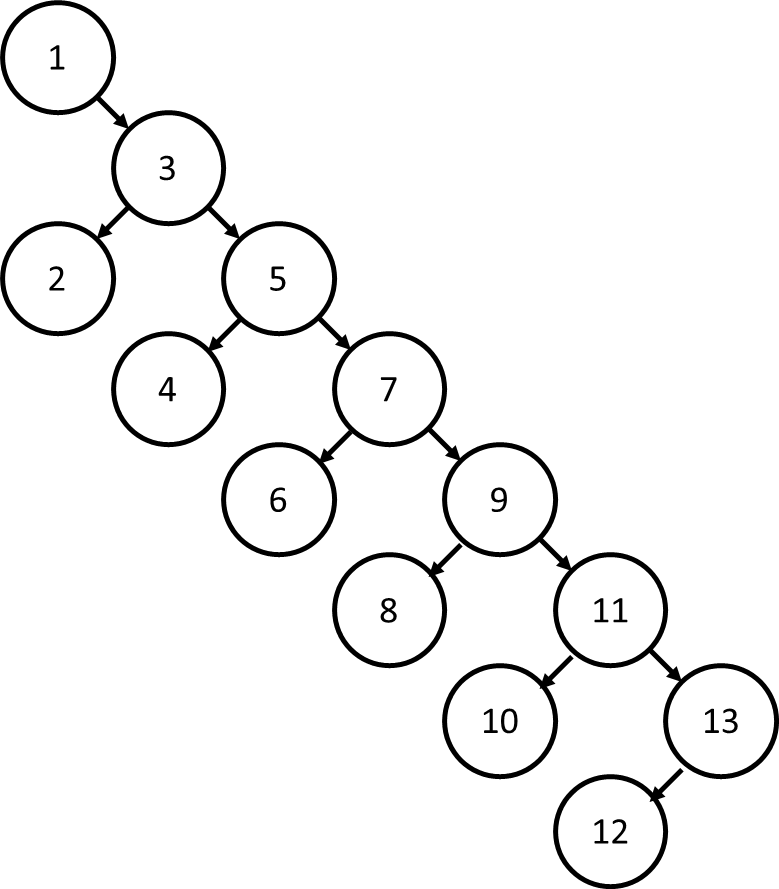
here **“n”** is the number of nodes.

but, the actual time complexity will deviate from the expected time complexity if the binary search tree is unbalanced.

in an unbalanced tree, it is possible to have a linked list as binary search tree, and the height of the tree can be up to n.

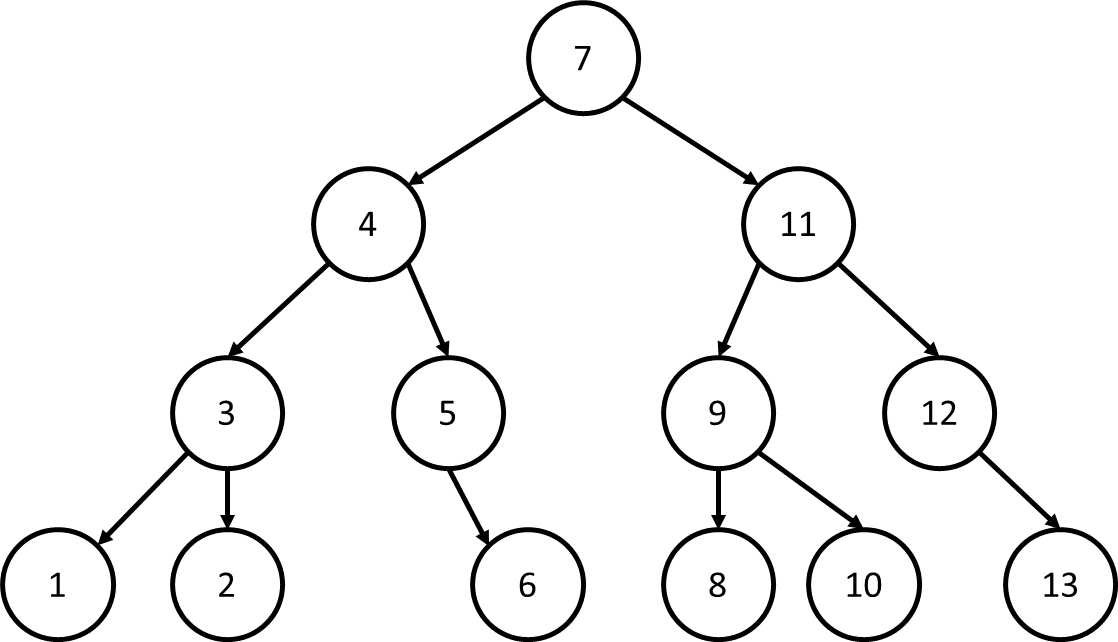
in this kind of cases, the actual complexities for all operations will be O(n).

3.



a) 7

b) , c)



minimum height = 3

insertion ordering = [7,4,11,3,5,9,12,1,2,6,8,10,13]