Opening Questions

1. In your own words explain how you insert a new key in a binary search tree.
2. Give an example of a series of 5 keys inserted one at a time into a binary search tree that will yield a tree of height 5.
3. If we do a left rotate on node X below, explain which left and/or right child links need to be changed.

```
      X
     / \
    a   Y
       / \
      b   c
```

Tree depth vs Tree Height

The length of the path from the root r to a node x is the depth of x in T. The height of a node in a tree is the number of edges on the longest simple downward path from the node to a leaf, and the height of a tree is the height of its root. The height of a tree is also equal to the largest depth of any node in the tree.

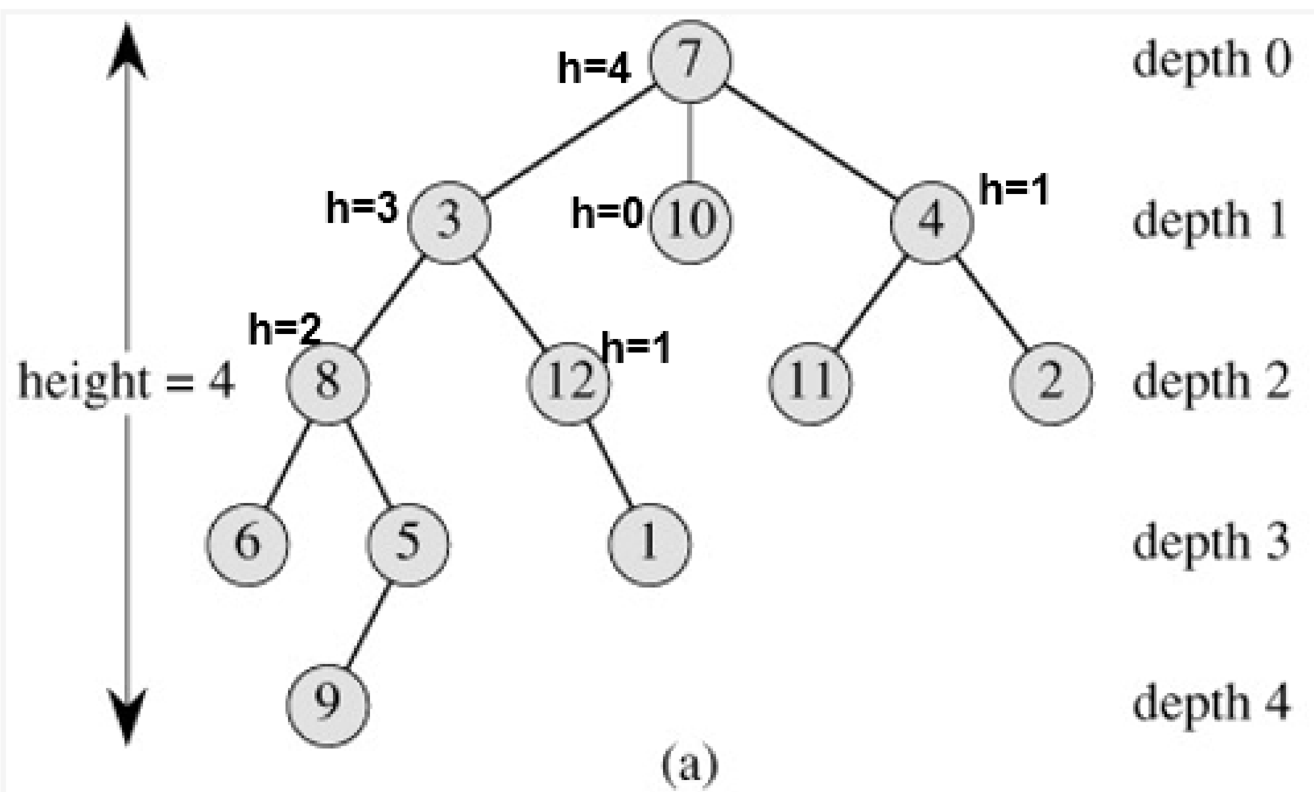
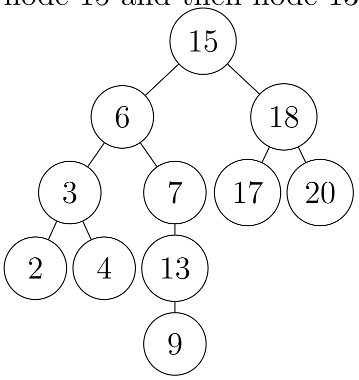


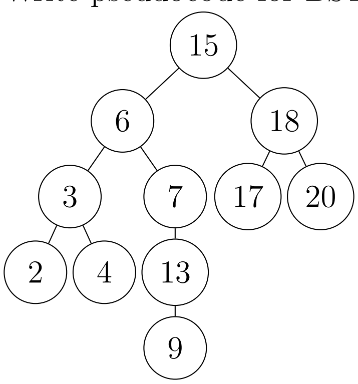
Figure 10.1: Height vs Depth of a Tree

BST Operations

1) Write pseudocode for BST Successor (or Predecessor). Demonstrate on the below tree from node 15 and then node 13.



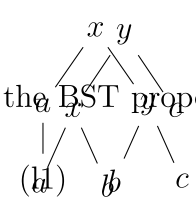
2) Write pseudocode for BST Insert. Demonstrate on the below tree to insert 5 and then 19.



3) What are the three possible cases when deleting a node from a BST?

4) Write pseudocode for BST Delete (assume you already have a pointer to the node)

BST Rotations



Local operation in a search tree that maintains the BST property and possibly alters the height of the BST.

x and y are nodes; a, b, c are sub trees

5. Write pseudocode for LeftRotate (or RightRotate). What is the worst case runtime?