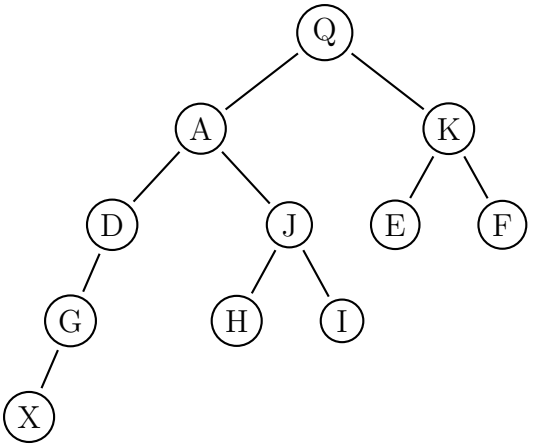
1. 
   - Traverse the binary tree using levelorder, preorder, inorder, and postorder:




   - What is the height of the above tree?
   - What is the height of node A? What is the height of node K?
   - What is the balance at A, D, and J?

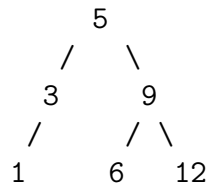
2. 
   - What is the maximum number of nodes in a tree with branching factor 2 and height $h$?
   - What is the minimum number of nodes in a tree with branching factor 2 and height $h$?
   - What is the maximum number of nodes in a tree with branching factor 3 and height $h$?
   - Approximately how may nodes are leaf nodes in a tree with branching factor 2 and height $h$?
   - Approximately how may nodes are leaf nodes in a tree with branching factor 3 and height $h$?
   - Given a binary tree with $N$ nodes, what is the maximum height of the tree?
   - Given a binary tree with $N$ nodes, what is the minimum height of the tree?

   Geometric progression sum formula may be useful:

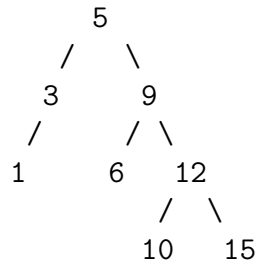
$$b^0 + b^1 + \ldots + b^m = \frac{b^{m+1}}{b-1}$$

3. Draw a parse tree for expression `*it = (a+b)*c+d/e;`

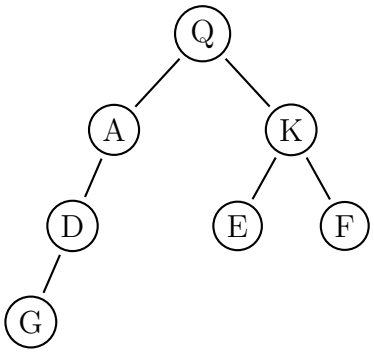
4. Insert into a Binary Search Tree below the following values: 4,7,15. Draw the final tree.




5. From the Binary Search Tree below delete values 3 and 9. Draw the final tree.

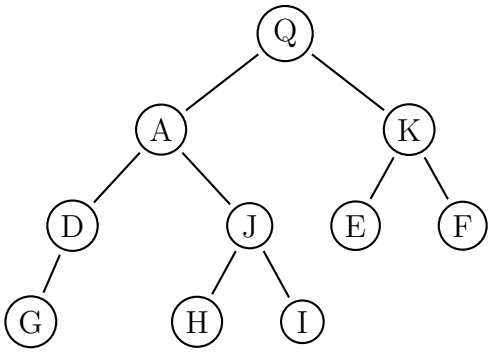



6. Given the tree below

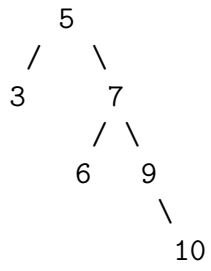



- Calculate balances in all nodes.
- Rotate left around Q. Calculate balances in all nodes.
- Rotate right around Q. Calculate balances in all nodes.

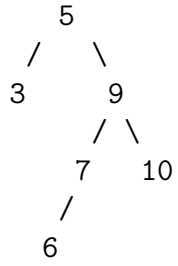
7. Given a **splay** tree below, assume node D is accessed. Draw the resulting tree.




8. Given the following tree: compute balance at each node. Show how to apply rotation(s) to make it an AVL tree, show all rotations if more then one.




9. Show how to apply rotation(s) to make it an AVL tree, show all rotations if more then one.



10. You are given a sequence of numbers that is a post-order traversal of an unknown BST. Reconstruct the tree. Understand logic, actual code may take some time.

    Execute on: {1,4,3,7,8,9,6}

11. You are given a sequence of numbers that is a level-order traversal of an unknown BST. Reconstruct the tree. Understand logic, actual code may take some time.

    Example: {5,3,7,6,9,10}. Consider: 5 must be the root, 3 is less then 5, so can be its left child, 7 is greater than 5, so can be its right child. Then: 6 cannot be left child of 3, also cannot be the right child of 3 (since it is greater than 5), so look at 7...

    Example: {5,2,1}. 5 is the root, 2 is its left child, 1 cannot be right child of 5, so go to the next level ...

    Do yourself: {7,2,9,5,8,4,6}

12. You are given a sequence of numbers that is a pre-order traversal of an unknown BST. Reconstruct the tree. Understand logic, actual code may take some time.

    Execute on: {3,1,2,8,6,5,7,9}

13. Will a problem similar to the previous 3 make sense for in-order traversal? Hint – look at the next problem.

14. Given a sorted array, write a function to build a **complete** binary search tree.

15. Given a BST, is there a pair of values with the sum of the 2 values a given number $S$. Write pseudo-code to solve problem efficiently. What is the run-time of your algorithm?