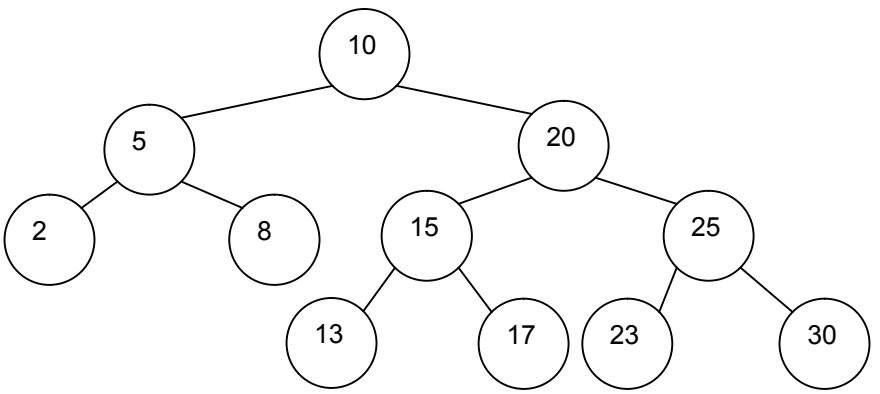
Section 4 Worksheet: AVL Trees & B-Trees

1. What 3 properties must an AVL tree have?

2. Draw an AVL tree of height 4 that contains the *minimum* possible number of nodes.

3. In a typical BST, inserting keys in order result in a worst-case height. Show the result when an initially empty AVL tree has keys 1 through 7 inserted in order.

4. For the following AVL tree:



   a. What values could you insert to cause a right-right imbalance, and at which node does the imbalance occur?
   b. How about a right-left imbalance? At which node does the imbalance occur?
   c. Insert 18 into the following AVL tree. What type of imbalance does it cause? Show the result after balancing.

5. AVL tree balance violation cases:
   a. Insert the following keys, in order, into an initially empty AVL tree: 12, 8, 9, 20, 10, 15, 3, 11, 5.
   b. Find a key we could insert into your resulting tree that would result in a case 1 balance violation (left-left).

6. Given a binary search tree, describe how you could convert it into an AVL tree with worst-case time O(nlogn) and best case O(n).

7. B-Trees:
   a. What constraints do the following values impose on a B-Tree: M=32, L=16?
   b. Insert the following into an empty B tree with M=3 and L=3: 12, 24, 36, 17, 18, 5, 22, 20.
   c. Delete 17, 12, 22, 5 & 36