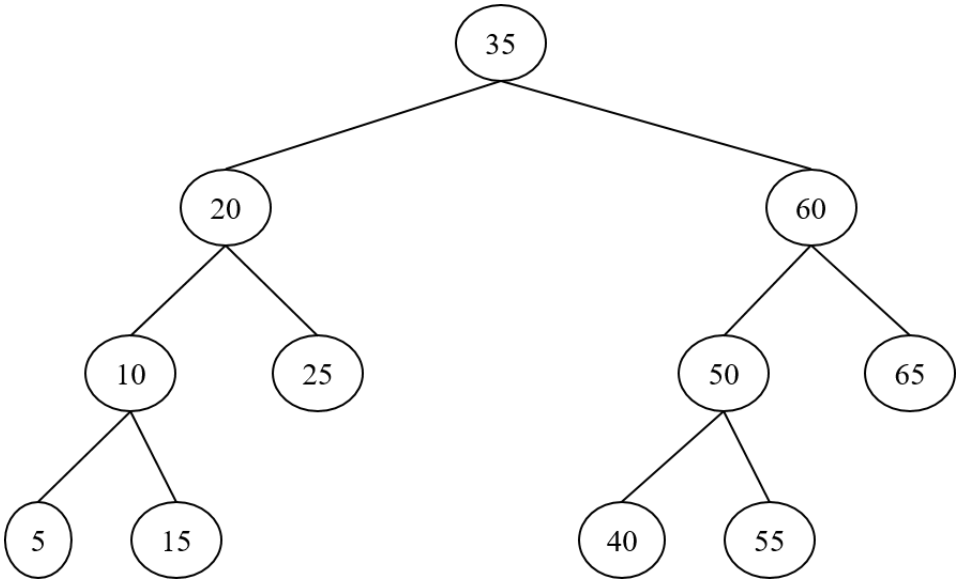
CPSC 131 Homework 7

Deadline: Wednesday, November 14 (Mon Wed sections)
Thursday, November 15 (Tue Thu sections)

Turn in your submission as a hard copy in class. Complete all homework problems. However, only some questions may be graded for credit.
Refer to your instructor's syllabus addendum to see their policy on group work. Some instructors allow homework to be completed in groups.

#1 [6 points]

Given the AVL tree below:



Answer the following questions (2 points each):

a. What is the height of each subtree in the tree? Write the height of each subtree and the balance factor next to each node. (similar to zyBook participation activity 7.1.1)

b. Draw the AVL tree after a new node with key "57" is inserted. Refer to the trinode restructuring handout.

c. Draw the AVL tree after the node with key "20" is deleted from the original tree (i.e., do not consider the previous insertion). Refer to the trinode restructuring handout.

#2 [4 points]

You are asked to create AVL trees with 7 keys: A, B, C, D, E, F, and G.

a. Draw the **shortest** AVL tree containing these 7 keys. (2 points)

b. Draw the **tallest** AVL tree containing these 7 keys. (There are multiple such trees, draw any one) (2 points)

#3 [4 points]

Consider the partial implementation of a Binary Search Tree class. For simplicity, each Node stores only the key. Add a public member function to class BST that returns the size of the tree (i.e., the number of the nodes).(Hint: think recursion!)

```
template <typename T>
class Node {
  T key;
  Node<T> *left, *right, *parent;
};

template <typename T>
class BST {
private:
  Node<T> root;
public:
  BST(): root(nullptr) {} // default constructor sets root to nullptr

  Node<T> *Search(const T &key) {
    Node<T> *cur = root;
    while (cur != nullptr)
      if (key == cur->key)
        return cur; // Found
      else if (key < cur->key)
        cur = cur->left;
      else
        cur = cur->right;
    return nullptr; // Not found
 }

 int size() {
   // YOUR CODE GOES HERE
 }
};
```

AVL Tree Trinode Restructuring