

CPSC 131 Homework 6

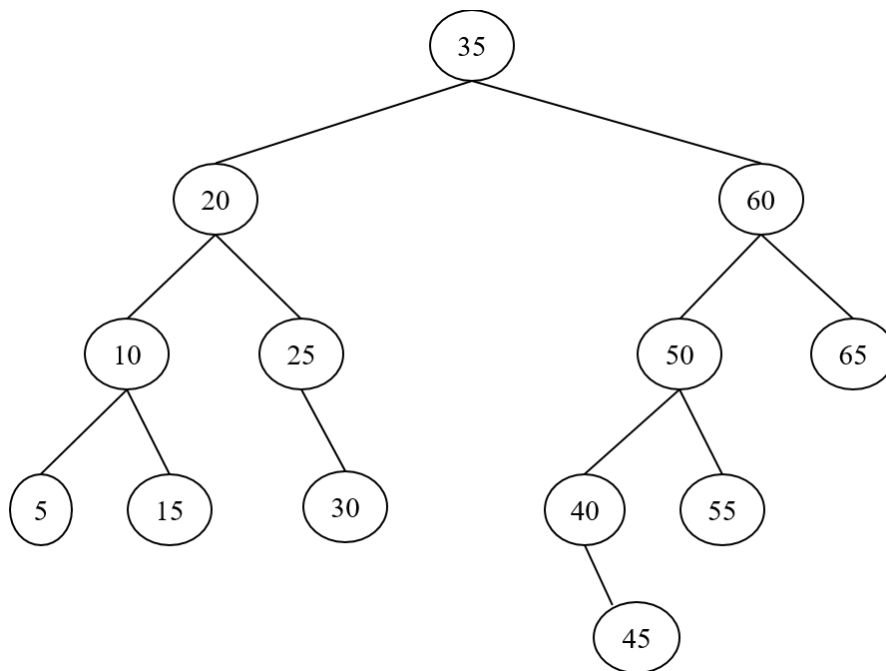
Deadline: Wednesday, November 7 (Mon Wed sections)
Thursday, November 8 (Tue Thu sections)

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 Binary Search Tree below:



Answer the following questions (1 point each):

- What is the depth of the node holding key 40?
- What is the height of the tree?
- List the nodes in the order of visit of an **inorder** traversal starting from the root node.

- d. Draw the tree after a new node with key "22" is inserted.
- e. Draw the original tree after the node with key "25" is deleted. (Do not consider the previous insertions or deletions.)
- f. Draw the original tree after the node with key "35" is deleted. (Do not consider the previous insertions or deletions.)

#2 [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 smallest value in the tree.

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

template <typename T>
class BST {
private:
    Node<T> root;
public:
    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
    }

    T getSmallest() {
        // YOUR CODE GOES HERE
    }
};
```