Name: Georges Hatem

CS 590 Homework 3: Binary Search Trees Reinforcement Exercises

Due Date: February 13, 2022

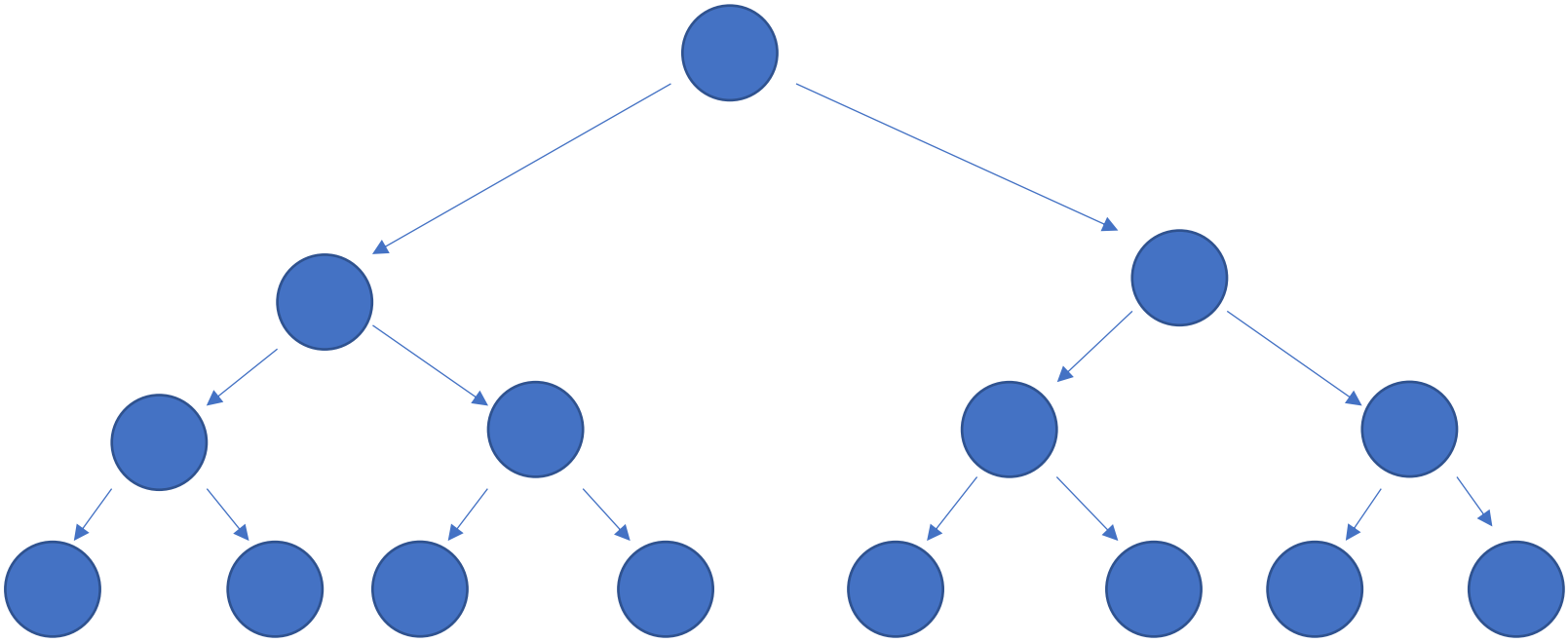
Problem 3.6.4:

To explain the reason why it is impossible for the sequence of number you encounter in the search described in this exercise statement to be (9, 12, 10, 11), I will start by constructing the Binary Search Tree (BST) based on what we are given in the exercise statement:

Two important factors (based on what stated in the exercise) to keep track of when constructing the Binary Search Tree (BST) are the height, and the numbers that are going to be in the tree:

First off, the numbers that are going to be in the tree are the integers from 1 to 15, inclusive. This means that we have 15 total numbers (n = 15).

A Binary Search Tree (BST) with a height of 4 can have a maximum of 15 numbers. This can be illustrated in the tree below:



This Binary Search Tree is of height 4 and contains a maximum of 15 nodes.

In order to structure our tree with height with the 15 numbers given, we must get the middle

number in the sequence of numbers given and put that number as the top (1st) node. This way we will have the rest 14 numbers distributed equally. The 7 numbers smaller than the middle number will be located in the 7 left nodes of the tree, and the 7 numbers greater than the middle number will be located in the 7 right nodes of the tree.

So, the middle in sequence integer numbers 1 to 15, inclusive is number 8. This means that 8 is the top (1st) node.

If we placed any other number in the top (1st) node, we will not obtain a tree with height of 4. We can start seeing that if we want to search for number 11, we have to pass first through 8 and not 9, which means that the sequence (9, 12, 10, 11) may be incorrect. Let's analyze this further to get to the result that when searching for the number 11, the sequence is not (9, 12, 10, 11):

Now, we know that numbers 1, 2, 3, 4, 5, 6, 7 should be located on left 7 nodes of the tree and numbers 9, 10, 11, 12, 13, 14, 15 should be located on the right 7 nodes of the tree.

Since the left and right children nodes of 8 (the top (1^{st}) node) are internal nodes and we have exactly 7 values to fit in there. Then we need to get the middle of both sequence.

The middle of sequence 1 to 7, inclusive is number 4, and the middle of sequence 9 to 15, inclusive is number 12.

This means that 4 and 12 are the left and right children of node 8 (top (1^{st}) node), respectively.

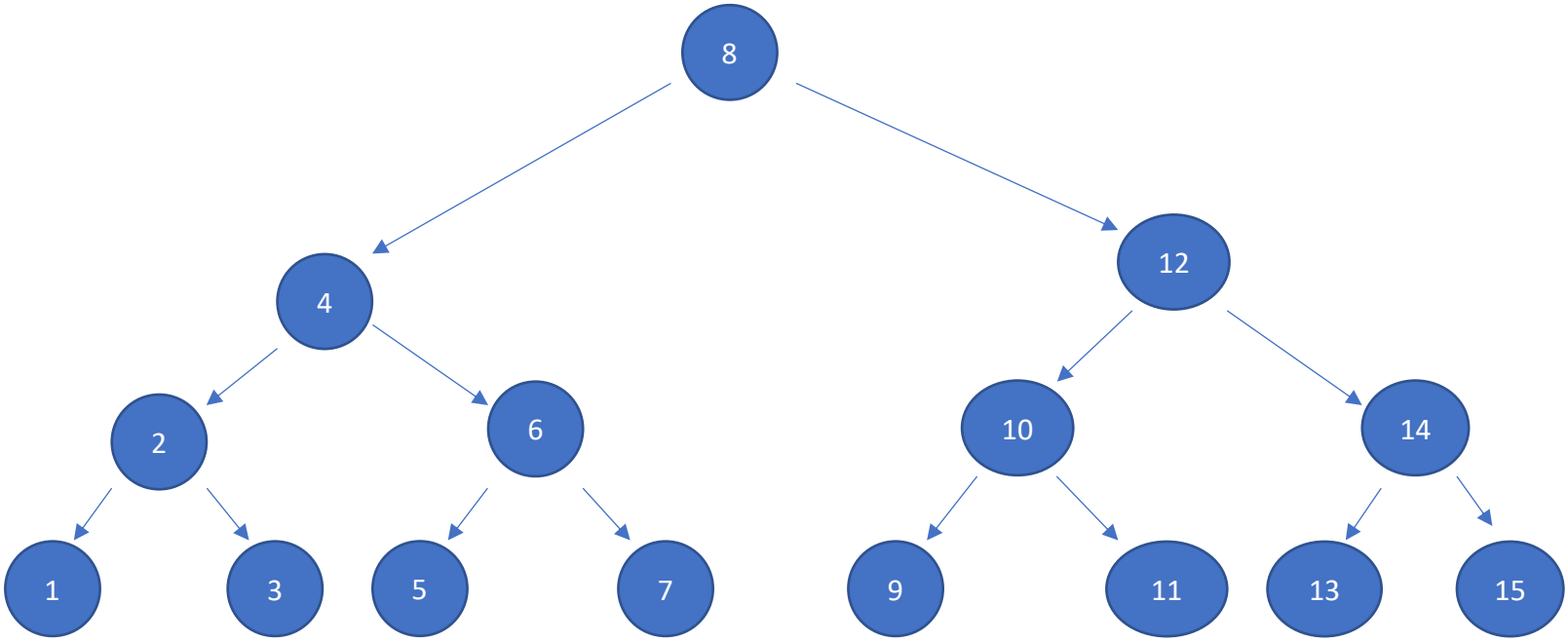
Now, we keep doing the same thing that we did on top. Numbers 1, 2, and 3 are less than 4 and therefore, should be located on the left side of node 4 and numbers 5, 6, 7 are greater than 4,

therefore, they should be located on the right side of node 4.

The middle of sequence 1, 2, and 3 is the number 2, and the middle of sequence 5, 6, and 7 is the number 6. This means that 2 and 6 are the left and right children of node 4, respectively.

Same analysis applies to the left and right children of node 12.

The Binary Search Tree (BST) that we obtain is as follows:

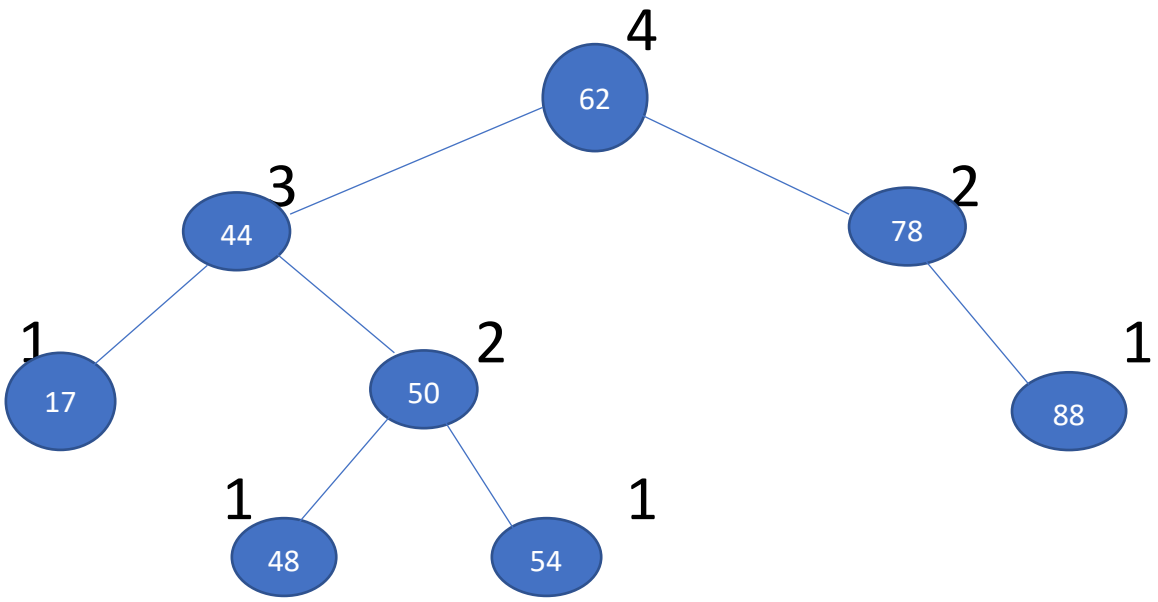


Based on the above Binary Search Tree, when searching for number 11, we pass through sequence (8, 12, 10, 11) and NOT (9, 12, 10, 11).

Therefore, it is impossible for the sequence of numbers I encounter in this search for number 11 to be (9, 12, 10, 11).

Problem 4.7.12:

Below is the AVL tree of Figure 4.3.3b:



Below is the Balanced AVL tree after removing item with key 62 from it: