Lab 07 Computer Science 1027B Summer 2025

Introduction

The purpose of these labs is to provide practice problems for you to get used to the concepts being covered in class. It is very important that you do these labs on your own without relying on available AI tools. Questions on exams are often very similar to exercises presented in these labs, so understanding them and solving these problems gives you a huge advantage for the exams. Make sure that you complete both the code as well as provide written answers to the blue questions within the boxes.

When you have concluded this lab, you should submit the following files to OWL Brightspace:

- Exercise_01.pdf (containing all relevant images)
- Exercise_02.txt (containing the answer to exercise 02)
- BinarySearchTree.java
- Exercise04.jpg/Exercise04.png/Exercise04.pdf (whatever file type you want)
- written_answers.txt (for the questions with blue font in the boxes)

Topics Covered

- Trees

Exercise 01

Given the following elements, draw the result of adding each element from left to right into a binary search tree. I recommend doing this on a sheet of paper to save time and then submitting a scan or photo of the paper.

29, 73, 17, 45, 43, 70, 12, 100, 88, 82

Q 01.1

What is the maximum number of nodes that could exist in a tree of this height?

Once each element has been added into the binary search tree, assume that we want to remove the root node (29). Draw both potential trees that could exist after this removal.

Q 01.2

Perform a **breadth-first search traversal** on this tree, printing the value of each node as you visit it. What is the resulting sequence of node values?

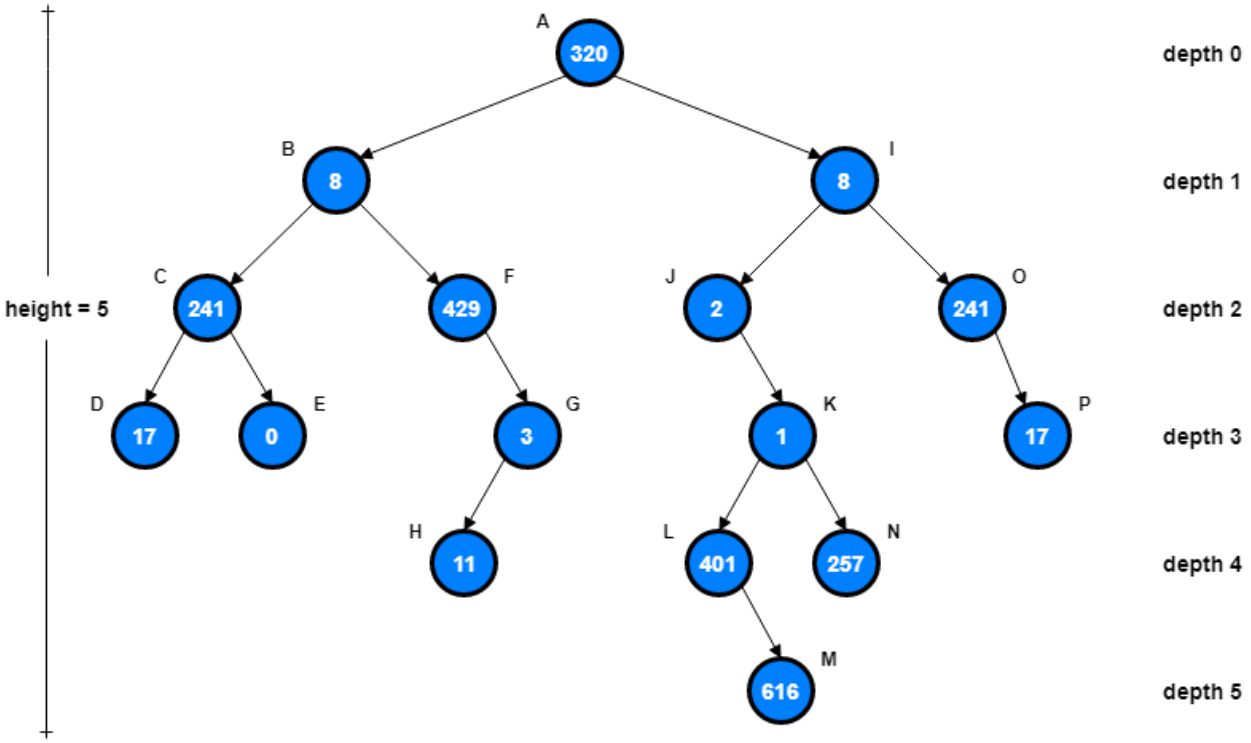
Q 01.3

Perform **depth-first search traversals** using the **preorder**, **inorder**, and **postorder** techniques on this tree. Print the value of each node as you visit it. What is the resulting sequence of node values for each?

Exercise 02

Determine whether the following tree is any of the following options, explaining why or why not.

1. Full?
2. Complete?
3. Balanced?
4. Perfect?
5. A binary search tree?
6. An AVL tree?



Q 02.1

What are three benefits of a BST over a non-BST?

Exercise 03

Given the following code, write a method, `private boolean containsRec(TreeNode root, int value)` that finds the given value in a BST.

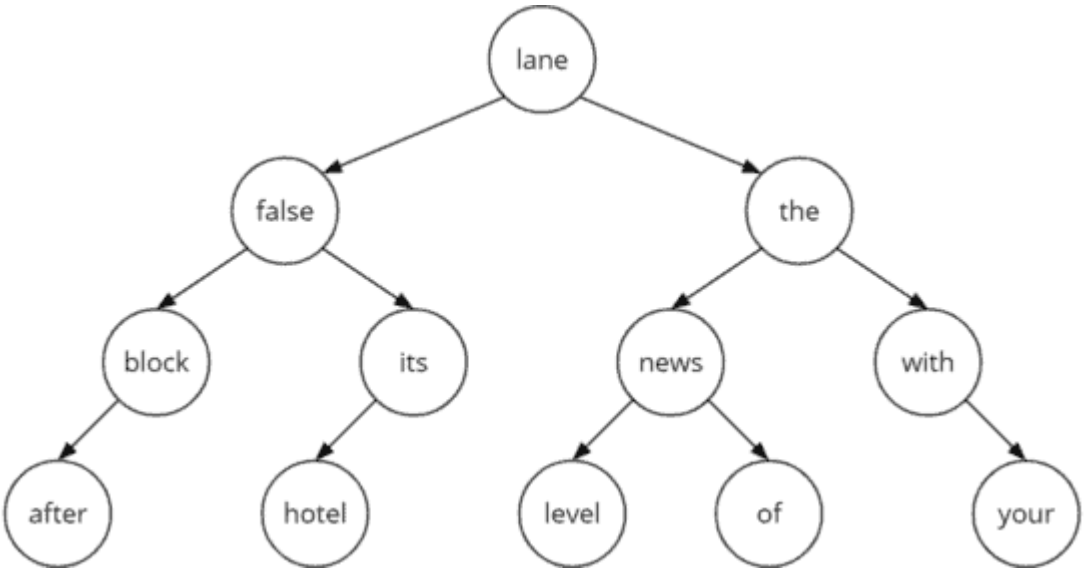
```
class TreeNode {
    int value;
    TreeNode left, right;

    TreeNode(int value) {
        this.value = value;
        left = right = null;
    }
}
```

```java
public class BinarySearchTree {
    TreeNode root;

    // method to insert a new node with a given value
    public void insert(int value) {
        root = insertRec(root, value);
    }

    // recursive function to insert a new value in the BST
    private TreeNode insertRec(TreeNode root, int value) {
        // If the tree is empty, return a new node
        if (root == null) {
            root = new TreeNode(value);
            return root;
        }

        // otherwise, recur down the tree
        if (value < root.value) {
            root.left = insertRec(root.left, value);
        } else if (value > root.value) {
            root.right = insertRec(root.right, value);
        }

        return root; // return the (unchanged) node pointer
    }

    // method to check if a certain value exists in the BST
    public boolean contains(int value) {
        return containsRec(root, value);
    }

    // recursive function to check if a certain value exists in the BST
    private boolean containsRec(TreeNode root, int value) {
        // your code here
    }

    public static void main(String[] args) {
        BinarySearchTree bst = new BinarySearchTree();

        // insert elements into the BST
        bst.insert(10);
        bst.insert(20);
        bst.insert(5);
        bst.insert(15);
        bst.insert(3);

        System.out.println("Does the BST contain 100? " + bst.contains(100)); // output: false
        System.out.println("Does the BST contain 3? " + bst.contains(3));   // output: true
    }
}
```

Lab 07 Computer Science 1027B Summer 2025

Exercise 04

Supposed we have the following BST



Remove the node containing ‘the’. Draw the new tree.

Q 04.1

What could be some potential real-world uses for BSTs where the nodes contain Strings?

Q 04.2

Could a BST contain nodes which hold objects of custom classes? If yes, what would you need to be sure to implement in the custom class?