

## CCCS 315: Data Structures and Algorithms

### Assignment #3

#### Type of Assignment

- *Individual Work*
- *Evaluated: this assignment is marked out of 100 points and is worth 10% of your final grade*
- *Estimated Time: 120 minutes*

#### Description

In this Assignment, you will study, Binary search tree, Avl trees and their benefits.

#### Learning outcomes being met through this assessment

- Recognize the use of the tree ADT
- Apply Big-O notation to analysis an algorithm
- Comprehending general searching algorithm.

#### Steps to complete the assignment

This assignment is designed similarly to a job interview. First, we will introduce a problem. You then need to apply critical thinking, design a solution, evaluate the solution (Big-O), then improve it if needed. Lastly, you will implement your solution.

#### Part 1: Search

Remember how in Assignment 1 you created a method to find the smallest value? Now we want to find a specific value, let's say for example 5 in an array or linked list.

- First create program that generates either a LinkedList or array of size  $n$  ( $=100$ ) [similar to Assignment\_1 or Assignment\_2]
  - Need a range for max and min
  - Int only
  - Define a `int myVal=5;` //this is going to be the value you are looking for
- Create Binary search tree and add the list or array to it,
- Define a search method (your call)
- Call it with either binary search ADT or the original LinkedList(or array) and return true if it has the value, and false if it doesn't. (**Do not use the build in methods that check if a linked list have it or not, I like to extend this homework later and we do need to find the value and have access to it!**) Note, in the future we can extend this class to be (key, value) pair and we search for key and return value, but for now we focus on just finding it.
- Like you have done in Assignment 2, Create a way to calculate the time.



- Create a method that loops through all the values and then find the item
- Now time both programs
- Analyze with Big-O the running time of both methods (one is your method, the other is just simple loop)

### Evaluation Criteria

- Format Requirements (10%)
  - Comments and CamelCase is important
  - Style
- Binary search tree (30%)
- Search method (20%)
- Looping method (10%)
- Big-O (30%)

### Submitting your Work

- You have to zip all the Java code for part 1 and part 2 (.java codes and if missing you will lose 85%)
- PDF for last part of part2

### Notes:

- Cheating or copying online without referencing correctly or explaining your code before and after online source has automatic failure as a grade.
  - If your references are not correct, you will lose marks.
  - (In Comments) Show your code before searching online AND showing what the issue was.
  - (In Comments) The reference.
  - (In Comments) How the online or book reference helped you change your code.
  - **Copying 80% from online with reference still results in ZERO!**
    - **Modify it to fit your code, do not blindly copy it and remember you are here to learn !**
- Do NOT share your code with anyone except your TA and me.
  - otherwise you will get a grade of ZERO.