M07-P04 – Functional paradigm: Racket

TIME ESTIMATE: 9 hours

DELIVERABLES: Submit your work on **Canvas**. Turn in your work before the specified deadline. You are expected to provide a working solution for each problem using Racket, Java, and C, and provide explanations of your code.

DEADLINE: Java Nov. 22th, C 25th, Racket Dec. 4th. After three days your work is not accepted. 10 points for every 24 hours after the deadline as penalty.

PROBLEM:
Implement the function for binary search tree.

A tree is a graph, G = {N, E}, where
1. Edges represent a parent-child relationship between nodes.
2. There is a single node, the root, having no parent.
3. All other nodes are involved in a parent-child relationship.
4. Nodes have exactly one parent, except the root node.
5. Nodes may have zero or more children.

Binary search tree
1. A binary search tree will have exactly two children, a left node and a right node. Note that the left node and/or right node might be null.
2. The left child of any given node will be smaller than the given node.
3. The right child of any given node will be greater than the given node.
4. Assume that the are not repeated elements in the tree.
5. Return true if the value exists in the tree, false otherwise.

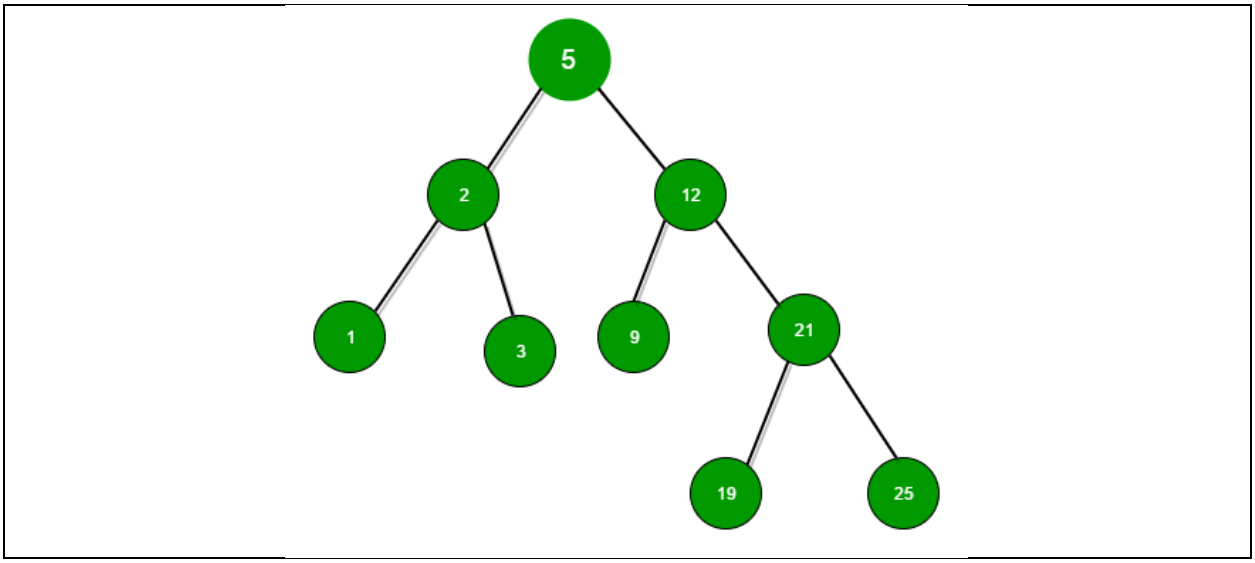

Figure 1: Example of a Binary Search Tree.
Taken from https://www.geeksforgeeks.org/largest-number-bst-less-equal-n/

TESTING: Your code will be tested against all possible test cases, plan accordingly. Failure to anticipate a test case will result in 2 points penalty.

Software Requirements:
R1. The software solution shall be implemented three times, in the following programming paradigms: object-oriented (P4.1), imperative (P4.2), and functional (P4.3)
R2. The software shall perform all possible test cases.
R3. For any given tree, T, the software shall return true or false when searching for a value.

GRADING:

Deliverable	P4.1	P4.2	P4.3
Program operates and produces some output.	10	10	20
Correct result on edge cases.	30	30	40
Code explanation provided at the beginning of your file.	10	10	10
Totals	50	50	70