CMPT 202
Quiz 3

April 8, 2019

Name:

1. Given n nodes, what is the height of a complete binary tree?
2. Prove that $f(n) = n + 1$ has big O complexity $O(n)$.
3. Prove that $f(n) = 100n$ has big O complexity $O(n)$
4. What is the big O complexity of binary search?
5. *Fill in the blank:* In a binary search tree, the ________ child is ________ than the root.
6. Is the following tree traversal recursive? If not, explain why. If so, label the base case and the recursive step(s)

```
public static void traverse(Node root) {
        if(root == null) {
                return;
        }
        else {
                traverse(root.leftChild);
                traverse(root.rightChild);
                System.out.println("Visited: " + root.content);
        }
}
```

7. *True or False:* The traversal in question 6 is a preorder traversal.
8. Linear probing and separate chaining are two methods to resolve collisions in hash tables. Explain how each resolves collisions.
9. Consider a complete binary search tree and a hash table that is almost empty. Also, assume that they contain the same data. Which one will give a faster look-up time? Would your answer change depending on the shape of the tree or how full the hash table is? Explain how.