COMP 352: Data Structures and Algorithms

Assignment 3

Summer 2020, sections AA

June 12, 2020

Shadi Jiha #40131284

Question 1:

1. Complexity:

public int depth(Node<T> root) {  
 if (root == null)  
 return 0;  
 else {  
  
 int left\_depth = depth(root.left());  
 int right\_depth = depth(root.right());  
  
 if (left\_depth > right\_depth)  
 return left\_depth + 1;  
 else  
 return right\_depth + 1;  
 }  
}

public void traversePreOrder(Consumer<Node<T>> operation) {  
 traversePreOrder(root, operation);  
}  
  
private void traversePreOrder(Node<T> root, Consumer<Node<T>> operation) {  
  
 if (root == null)  
 return;  
  
 operation.accept(root);  
 traversePreOrder(root.left(), operation);  
 traversePreOrder(root.right(), operation);  
}

Then inside main we can call:

List<Integer> depth\_of\_nodes = new ArrayList<>();  
tree.traverseInOrder(e -> depth\_of\_nodes.add(tree.depth(e)));

1. Complexity:

public static int Count\_Full\_Nodes(BinarySearchTree<?> tree) {  
  
 AtomicInteger count = new AtomicInteger();  
 tree.traverseInOrder(e -> {  
 if (e.hasLeft() && e.hasRight())  
 count.getAndIncrement();  
 });  
  
 return count.get();  
}

**Question 2:**

**Question 3:**

1. [195=0, 91=0], [16=0, 94=0, 81=0], [147=0], [265=0], [32=0, 162=0], [189=0, 202=0], [21=0], [48=0], [75=0], [180=0, 37=0], [207=0, 77=0]
2. 7 Collisions

**Question 4:**

No, it doesn’t. The collision number went from 7 to 8.