CS 1332 Recitation 4 Worksheet — BST Traversals, Iteratable/Iterator, Heaps Intro

CS 1332 TAs

February 5, 2019

This worksheet covers material from this week's recitation. It is meant to be additional exercise for your benefit and will not be graded. Feel free to collaborate with other students on this and ask TAs for help. Distribution of this worksheet is **not** permitted.

1 Questions

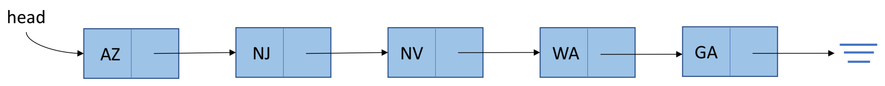
1. Runtime of iterating through a `LinkedList` with an `Iterator`: ______
2. T/F Every `Iterator` must implement the `next()` method: ______
3. T/F Every `Iterator` must implement the `remove()` method: ______
4. Runtime of computing a pre-order traversal: ______
5. Runtime of computing a in-order traversal: ______
6. Runtime of computing a post-order traversal: ______
7. What data structure is used to compute the level-order traversal of a tree?: ______

2 Tracing

Given the following `Iterator` code that iterates over the given linked list `list`, draw what the list `output` would look like following the execution.

```
LinkedList<String> list = new LinkedList<>();
Iterator<String> it = list.iterator();
while (it.hasNext()) {
    String s = it.next();
    if (s.charAt(0) == 'N') {
        output.addLast(s);
    }
}
```

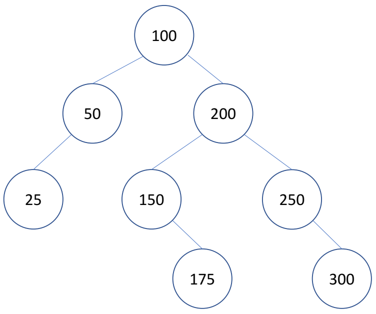
`list =`



```
output =
```

3 Diagramming

1. Find the pre-order, in-order, post-order, and in-order traversal for the following tree.



4 Above and Beyond

This is a small set of challenge questions related to the material in the course, similar to those that you may find in a Software Engineering interview. Again, feel free to collaborate with other students or ask a TA for help.

1. Write a function that determines finds the *lowest common ancestor (LCA)* of two nodes in a given BST. The LCA of two nodes is the lowest vertex (in terms of depth) where both nodes are descendants.

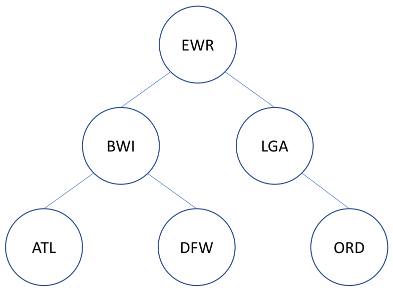
   Example:
   Input: Tree from problem Sec. 2 problem 2, nodes 175 and 300
   Output: The LCA of 175 and 300 is node 200
   Expected time: $O(n)$, Expected space: $O(1)$ outside of recursive stack

```
public Node lowestCommonAncestor(Node root, Node p, Node q) {

}
```

2. Given a BST, determine what the kth smallest element in the BST is for a given k

   Examples:
   Input: $k = 2$



   Output: BWI

   Expected time: $O(n)$, Expected space: $O(1)$ outside of recursive stack

```
public boolean kthSmallest(Node root, int k) {

}
```