

Name: Kyla Wilson
J-Number: J00813814

1.

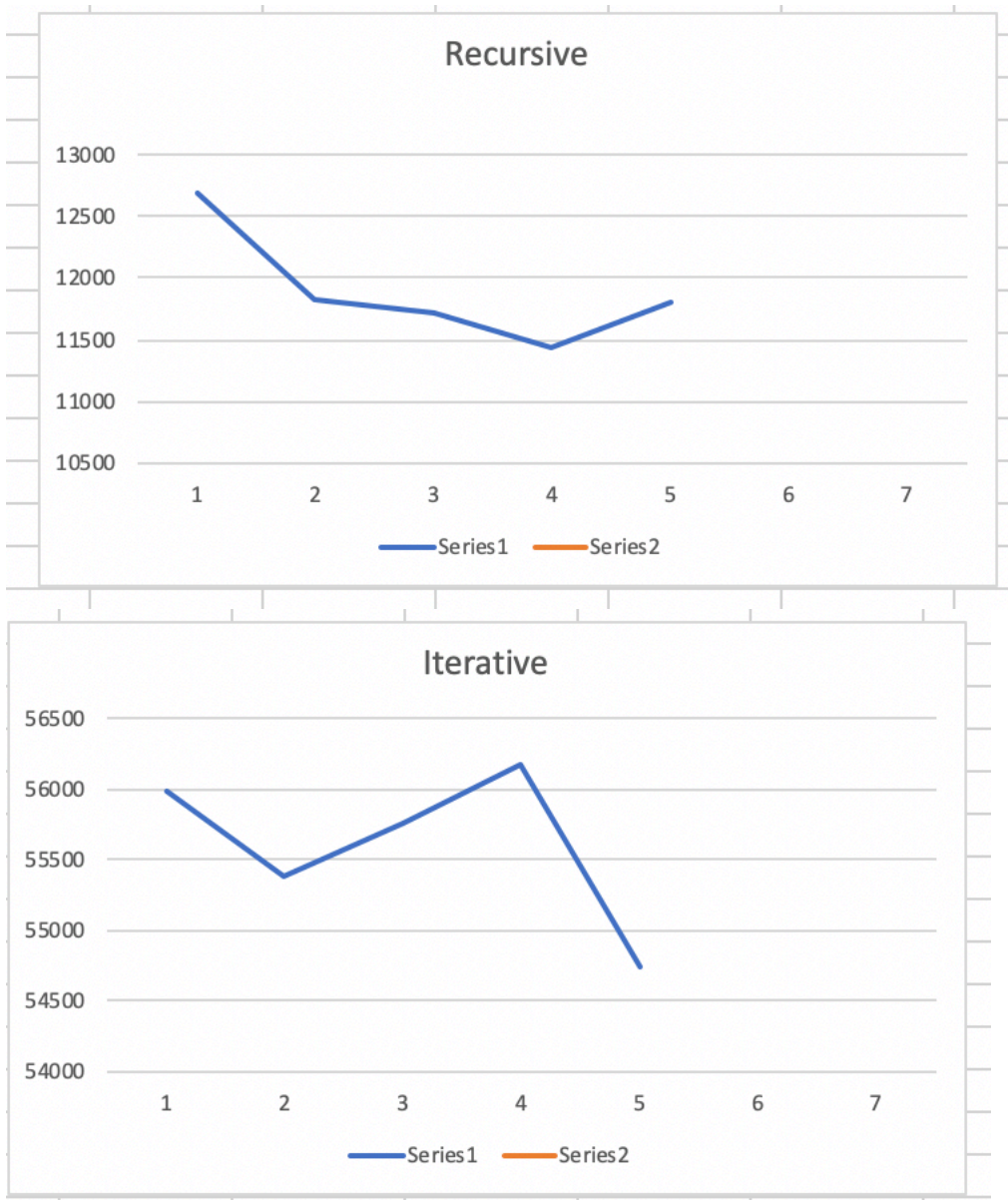
Stack nodesOnRight;

BTNode currentNode = TreeNodes[0]

```
While(currentNode IS NOT NULL) {  
    if(currentNode HAS RIGHT CHILD) {  
        nodesOnRight.push(RIGHT CHILD ID) O(1)  
    }  
  
    currentNode = currentNode LEFT CHILD PTR  
  
    if(currentNode IS NULL) {  
        if(nodesOnRight IS NOT EMPTY)  
            currentNode = TreeNodes[nodesOnRight.pop] O(1)  
    }  
}
```

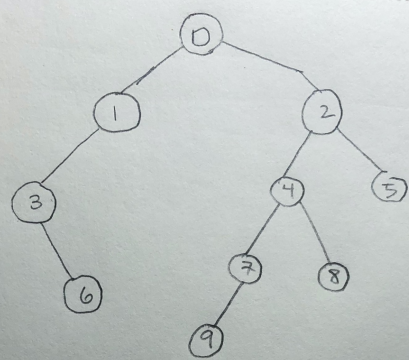
Assuming the array of nodes is not empty, the time complexity of this algorithm is $O(n)$ because it has to transverse through the entire list. Therefore, the best and worst case will be the same. $\Theta(n)$

```
Enter the file name for the edges of the tree:  
/Users/incrediblekyla/Documents/School/Spring2020/Datastructures&Algorithms/Homeworks/Datastructures&Algorithms/Datastructures&Algorithms/Assignment9/binaryTree-2000001.txt  
Enter number of nodes: 2000001  
Processing time (microseconds, recursive): 11815.3  
Processing time (microseconds, iterative): 54737.3  
Program ended with exit code: 0  
  
Enter the file name for the edges of the tree:  
/Users/incrediblekyla/Documents/School/Spring2020/Datastructures&Algorithms/Homeworks/Datastructures&Algorithms/Datastructures&Algorithms/Assignment9/binaryTree-2000001.txt  
Enter number of nodes: 2000001  
Processing time (microseconds, recursive): 11817.9  
Processing time (microseconds, iterative): 55373.3  
Program ended with exit code: 0  
  
Enter the file name for the edges of the tree:  
/Users/incrediblekyla/Documents/School/Spring2020/Datastructures&Algorithms/Homeworks/Datastructures&Algorithms/Datastructures&Algorithms/Assignment9/binaryTree-2000001.txt  
Enter number of nodes: 2000001  
Processing time (microseconds, recursive): 11728  
Processing time (microseconds, iterative): 55767.5  
Program ended with exit code: 0  
  
Enter the file name for the edges of the tree:  
/Users/incrediblekyla/Documents/School/Spring2020/Datastructures&Algorithms/Homeworks/Datastructures&Algorithms/Datastructures&Algorithms/Assignment9/binaryTree-2000001.txt  
Enter number of nodes: 2000001  
Processing time (microseconds, recursive): 11433.3  
Processing time (microseconds, iterative): 56170.3  
Program ended with exit code: 0  
  
Enter the file name for the edges of the tree:  
/Users/incrediblekyla/Documents/School/Spring2020/Datastructures&Algorithms/Homeworks/Datastructures&Algorithms/Datastructures&Algorithms/Assignment9/binaryTree-2000001.txt  
Enter number of nodes: 2000001  
Processing time (microseconds, recursive): 11817.9  
Processing time (microseconds, iterative): 55373.3  
Program ended with exit code: 0
```



The recursive function is faster because it is less operations...

Loop: 2
 Current Node:
 Stack:



Loop	0	1	2	3	4	5	6	7	8	9
Current Node	0	1	3	6	2	4	7	9	8	5
Stack	2	2	2, 6	2	5	5, 8	5, 8	5, 8	5	Empty

Preorder = 0, 1, 3, 6, 2, 4, 7, 9, 8, 5