Michael Schloss

CIS 3501

10/10/2019

**Big O**

1. Struct node O(1)
2. struct node\* newNode(int inNodeValue) O(1)
3. node\* CreateNewNode(int nodeValue) O(1)
4. node\* BuildAtree(node\* currNode, int nodeValue) O(n)
5. bool IsTreeBinary(node\* root, int nodeValue) O(1)
6. void PrintInOrder(node\* currNode, int nodeValue) O(n)
7. void PrintPreOrder(node\* currNode, int nodeValue) O(n)
8. void PrintPostOrder(node\* currNode, int nodeValue) O(n)
9. int main() O(n)

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | List of Words | Expected Output | Actual Output |
| 1 | 22  10  35  6  17  -1  43  3  -1  -1  -1  -1  -1  -1  -1  -1  -1 | This tree is binary  Pre-Order Traversal: 22 10 6 3 17 35 43  In-Order Traversal: 3 6 10 17 22 35 43  Post-Order Traversal: 3 6 17 10 43 35 22 | This tree is binary  Pre-Order Traversal: 22 10 6 3 17 35 43  In-Order Traversal: 3 6 10 17 22 35 43  Post-Order Traversal: 3 6 17 10 43 35 22 |
| 2 | 22  10  35  6  17  43  3 | This tree is binary  Pre-Order Traversal: 22 10 6 3 17 35 43  In-Order Traversal: 3 6 10 17 22 35 43  Post-Order Traversal: 3 6 17 10 43 35 22 | This tree is binary  Pre-Order Traversal: 22 10 6 3 17 35 43  In-Order Traversal: 3 6 10 17 22 35 43  Post-Order Traversal: 3 6 17 10 43 35 22 |
| 3 | 8  3  10  1  6  17  4  7  13 | This is a binary tree  Pre-Order Traversal: 8 3 1 6 4 7 10 17 13  In-Order Traversal: 1 3 4 6 7 8 10 13 17  Post-Order Traversal: 1 4 7 6 3 13 17 10 8 | This is a binary tree  Pre-Order Traversal: 8 3 1 6 4 7 10 17 13  In-Order Traversal: 1 3 4 6 7 8 10 13 17  Post-Order Traversal: 1 4 7 6 3 13 17 10 8 |
| 4 | -1 | This is not a binary tree  Pre-Order Traversal: n/a  In-Order Traversal: n/a  Post-Order Traversal: n/a | This is not a binary tree  Pre-Order Traversal: n/a  In-Order Traversal: n/a  Post-Order Traversal: n/a |