Problem: Implement depth-first traversals (in-order, pre-order, post-order) on a binary tree.

Constraints

- **Can we assume we already have a Node class with an insert method?**
    - **Yes**
- **What should we do with each node when we process it?**
    - **Call an input method visit_func on the node**
- **Can we assume this fits in memory?**
    - **Yes**

Test Cases

In-Order Traversal

- **5, 2, 8, 1, 3 -> 1, 2, 3, 5, 8**
- **1, 2, 3, 4, 5 -> 1, 2, 3, 4, 5**

Pre-Order Traversal

- **5, 2, 8, 1, 3 -> 5, 2, 1, 3, 8**
- **1, 2, 3, 4, 5 -> 1, 2, 3, 4, 5**

Post-Order Traversal

- **5, 2, 8, 1, 3 -> 1, 3, 2, 8, 5**
- **1, 2, 3, 4, 5 -> 5, 4, 3, 2, 1**

Algorithm

Test Cases

Note:

- **This following are all forms of depth-first traversals**

In-Order Traversal

- **Recursively call in-order traversal on the left child**
- **Visit the current node**
- **Recursively call in-order traversal on the right child**

Complexity:

- **Time: O(n)**
- **Space: O(m), where m is the recursion depth, or O(1) if using an iterative approach**

Pre-Order Traversal

- **Visit the current node**
- **Recursively call pre-order traversal on the left child**
- **Recursively call pre-order traversal on the right child**

Complexity:

- **Time: O(n)**
- **Space: O(m), where m is the recursion depth, or O(1) if using an iterative approach**

Post-Order Traversal

- **Recursively call post-order traversal on the left child**
- **Recursively call post-order traversal on the right child**
- **Visit the current node**

Complexity:

- **Time: O(n)**
- **Space: O(m), where m is the recursion depth, or O(1) if using an iterative approach**