Morris Traversal

Table of Contents

Introduction

Morris (InOrder) traversal is a tree traversal algorithm that does not employ the use of recursion or a stack. In this traversal, links are created as successors and nodes are printed using these links. Finally, the changes are reverted back to restore the original tree.

Algorithm

- Initialize the root as the current node curr.
- While curr is not NULL, check if curr has a left child.
- If curr does not have a left child, print curr and update it to point to the node on the right of curr.
- Else, make curr the right child of the rightmost node in curr's left subtree.
- Update curr to this left node.

Code

```
void Morris(struct Node* root)
{
	struct Node *curr, *prev;
	if (root == NULL)
		return;
	curr = root;
	while (curr != NULL) {
		if (curr->left_node == NULL) {
			cout << curr->data << endl;
			curr = curr->right_node;
		}
		else {
			/* Find the previous (prev) of curr */
			prev = curr->left_node;
			while (prev->right_node != NULL && prev->right_node != curr)
				prev = prev->right_node;

			/* Make curr as the right child of its previous */
			if (prev->right_node == NULL) {
				prev->right_node = curr;
				curr = curr->left_node;
			}

			/* fix the right child of previous */
			else {
				prev->right_node = NULL;
				cout << curr->data << endl;
				curr = curr->right_node;
			}
		}
	}
}
```