Discuss an algorithm to traverse a tree. Using Depth first algorithm.

Dept-first Search is an algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

There are three different types of depth-first traversals:

1. **PreOrder traversal:** This is achieved by visiting the parent node first, followed by left child node and right child node
2. **InOrder traversal:** This is achieved by visiting the left child node first, followed by parent node and right child node
3. **PreOrder traversal:** This is achieved by visiting the left child node first, followed by right child node and parent node

The following is an algorithm for PreOrder traversal

**Step 1:** Add a node to the stack

**Step2:** Pop out an element from the stack and add its right and left children to stack

**Step 3:** Pop out an element and print it and add its children

**Step 4:** Repeat step 3 and 3 until the Stack is empty

The following Java code implements the algorithm.

public class BSTDFS {

public void DFS(Node root) {

Stack<Node> s = new Stack<Node>();

s.add(root);

while (s.isEmpty() == false) {

Node x = s.pop();

if(x.right!=null) s.add(x.right);

if(x.left!=null) s.add(x.left);

System.out.print(" " + x.data);

}

}

public static void main(String args[]){

Node root = new Node(1);

root.left = new Node(2);

root.left.left = new Node(4);

root.left.right = new Node(5);

root.right = new Node(3);

root.right.left = new Node(6);

root.right.right = new Node(7);

BSTDFS b = new BSTDFS();

System.out.println("Depth-First-Search : ");

b.DFS(root);

}

}

class Node {

int data;

Node left;

Node right;

public Node(int data) {

this.data = data;

left = null;

right = null;

}

}

**The ouput of the above implementation is:**  1 2 4 5 3 6