Data Structures (15B11CI311)

Odd Semester 2020

3rd Semester , Computer Science and Engineering

Jaypee Institute Of Information Technology (JIIT), Noida

Lecture: 29

Topics to be covered:

- Threaded Binary Tree
- Threaded Binary Search Tree
- Traversals in Threaded Tree

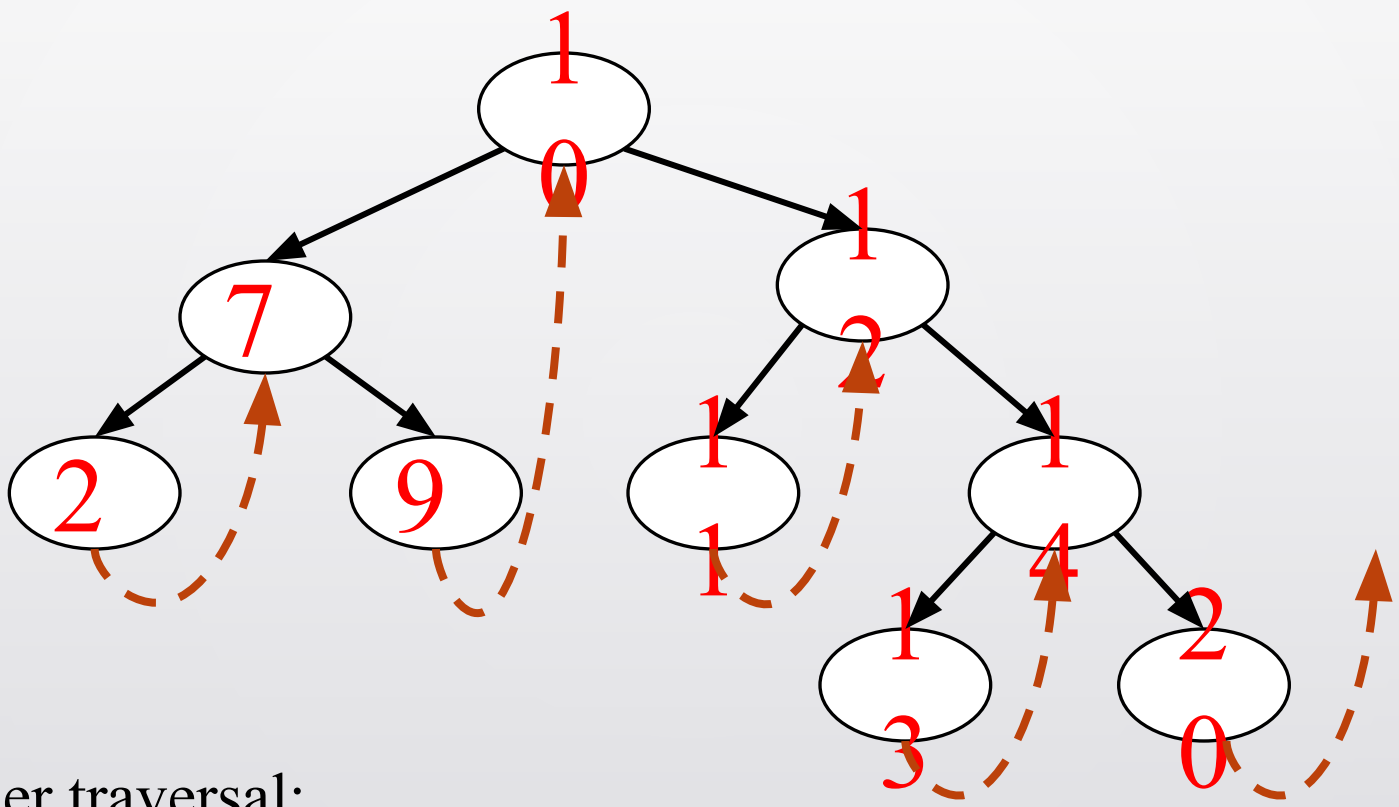
Threaded Trees

- Binary trees have many NULL pointers which are the wasted space
  - Number of nodes in a binary tree = n
  - Number of non-NULL pointers = n-1
  - Total pointers = 2n
  - NULL Pointers = 2n-(n-1) = n+1
- These NULL pointers can be replaced with some useful threads.
- If left child of a node is NULL then replace it with a pointer (thread) to the node that comes before that node in an in-order traversal (in-order predecessor)
- If right child of a node is NULL then replace it with a pointer (thread) to the node that comes after that node in an in-order traversal (in-order successor)

Threaded Trees

- Types of Threaded Binary Trees
  - Single Threaded: Only single type of thread is present, i.e. either towards in-order predecessor or in-order successor.
  - Double threaded: Both threads are presented, i.e. towards both the in-order predecessor and in-order successor.
- Threaded Binary Trees can be used for faster in-order traversals as it will save the required space for recursion in normal binary tree.

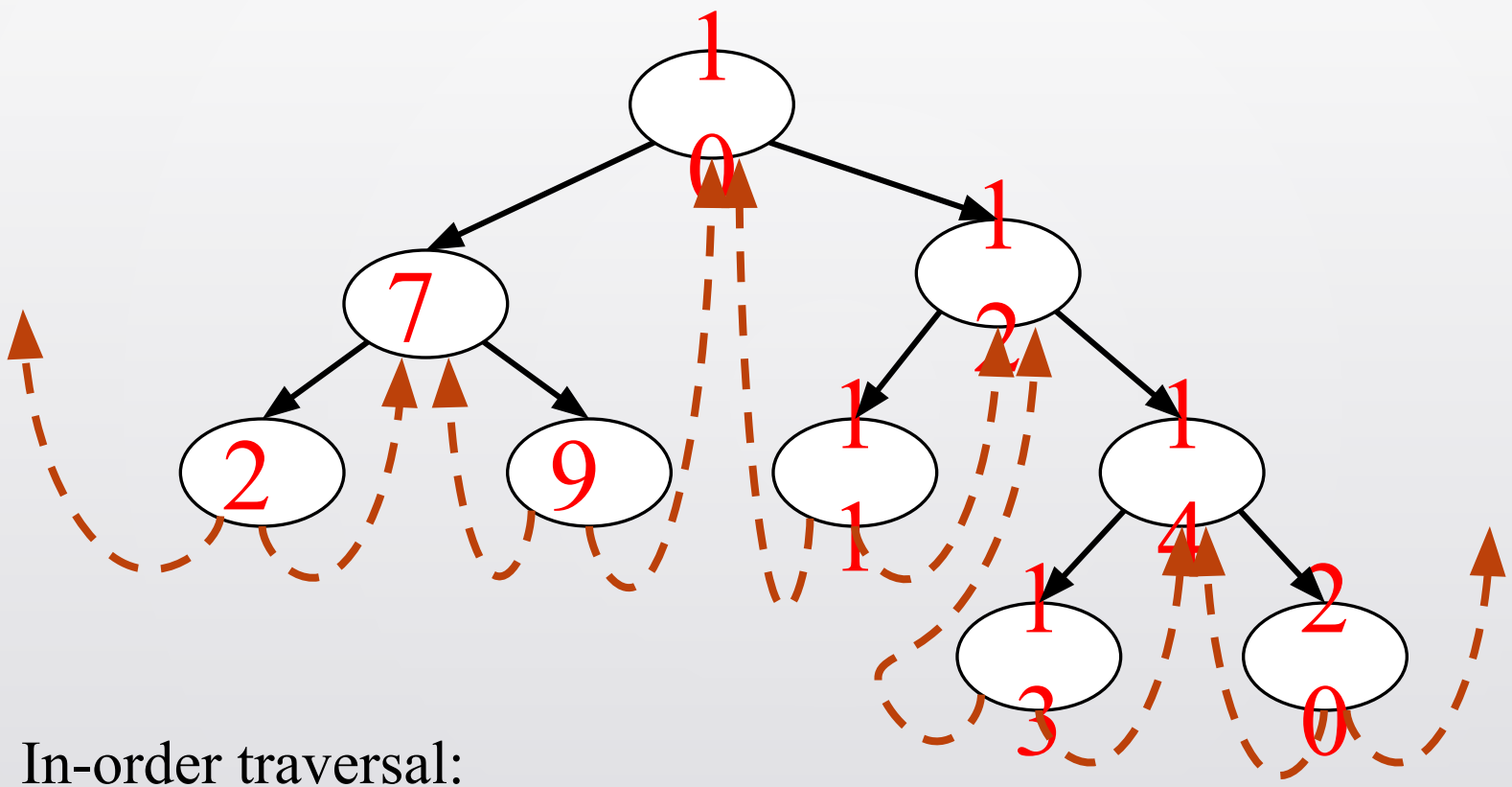
Single Threaded Binary Search Tree Example



In-order traversal:

2, 7, 9, 10, 11, 12, 13, 14, 20

Double Threaded Binary Search Tree Example



In-order traversal:

2, 7, 9, 10, 11, 12, 13, 14, 20

Representation of Threaded Binary Tree

- Each node of a threaded binary tree contains two extra information
  - Left thread field
  - Right thread field
  - The left and right thread fields of a node can have two values:
    - 1: Indicates a normal link to the child node
    - 0: Indicates a thread pointing to the in-order predecessor or in-order successor

```
struct Node {
    int data;
    Node *lchild, *rchild;
    bool lthread, rthread; };
```

Threaded Binary Tree In-order Traversal

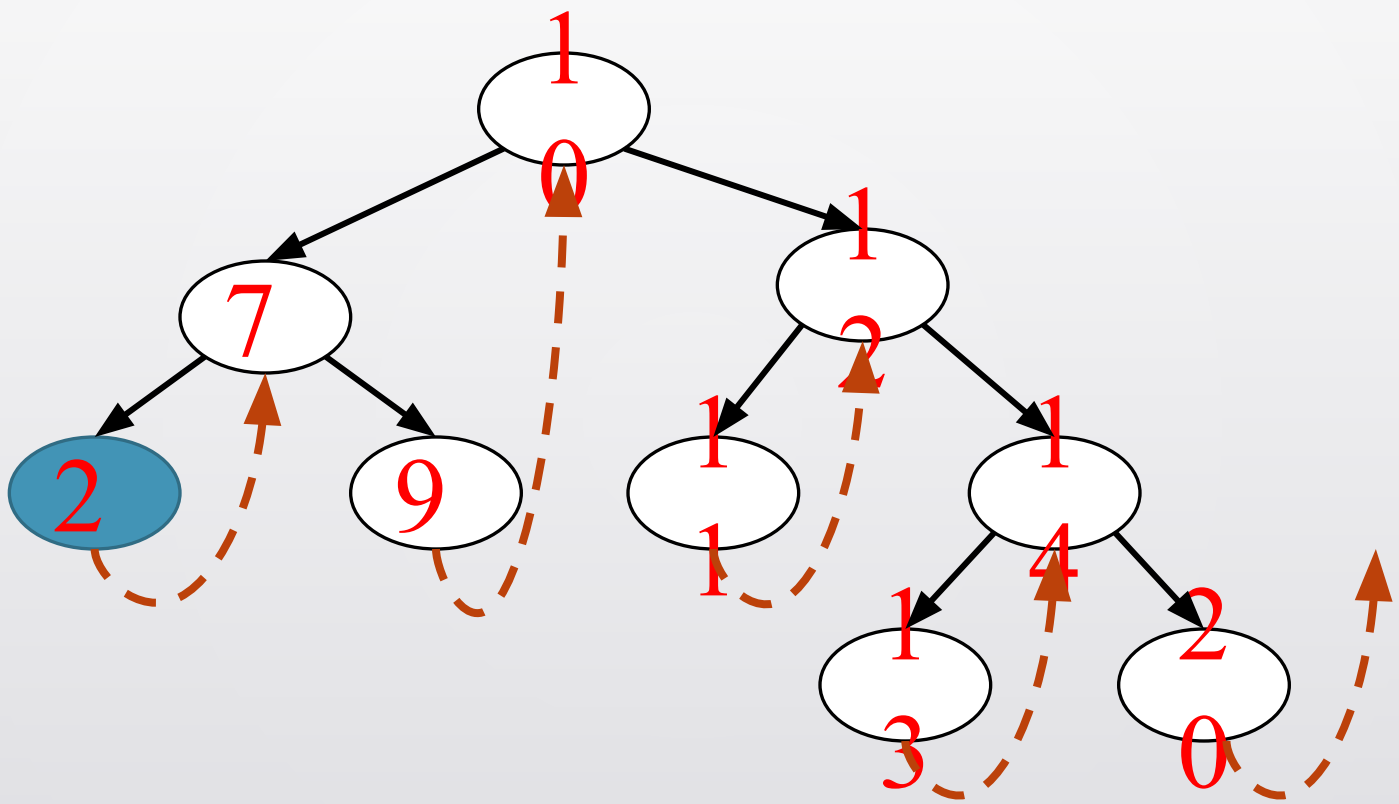
```
void inorder(Node *root) {
   Node *cur = leftmost(root);
    while (cur != NULL)
    {
        cout<< cur->data;
        if (cur->rthread)
            cur = cur->rchild;
        else
      cur = leftmost(cur->rchild);
    }
}
```

```
Node* leftMost(Node *n) {
    if (n == NULL)
        return NULL;

    while (n->lchild != NULL)
        n = n->lchild;
    return n;
}
```

Source: https://www.geeksforgeeks.org/threaded-binary-tree/#:~:text=The%20idea%20of%20threaded%20binary,types%20of%20threaded%20binary%20trees.

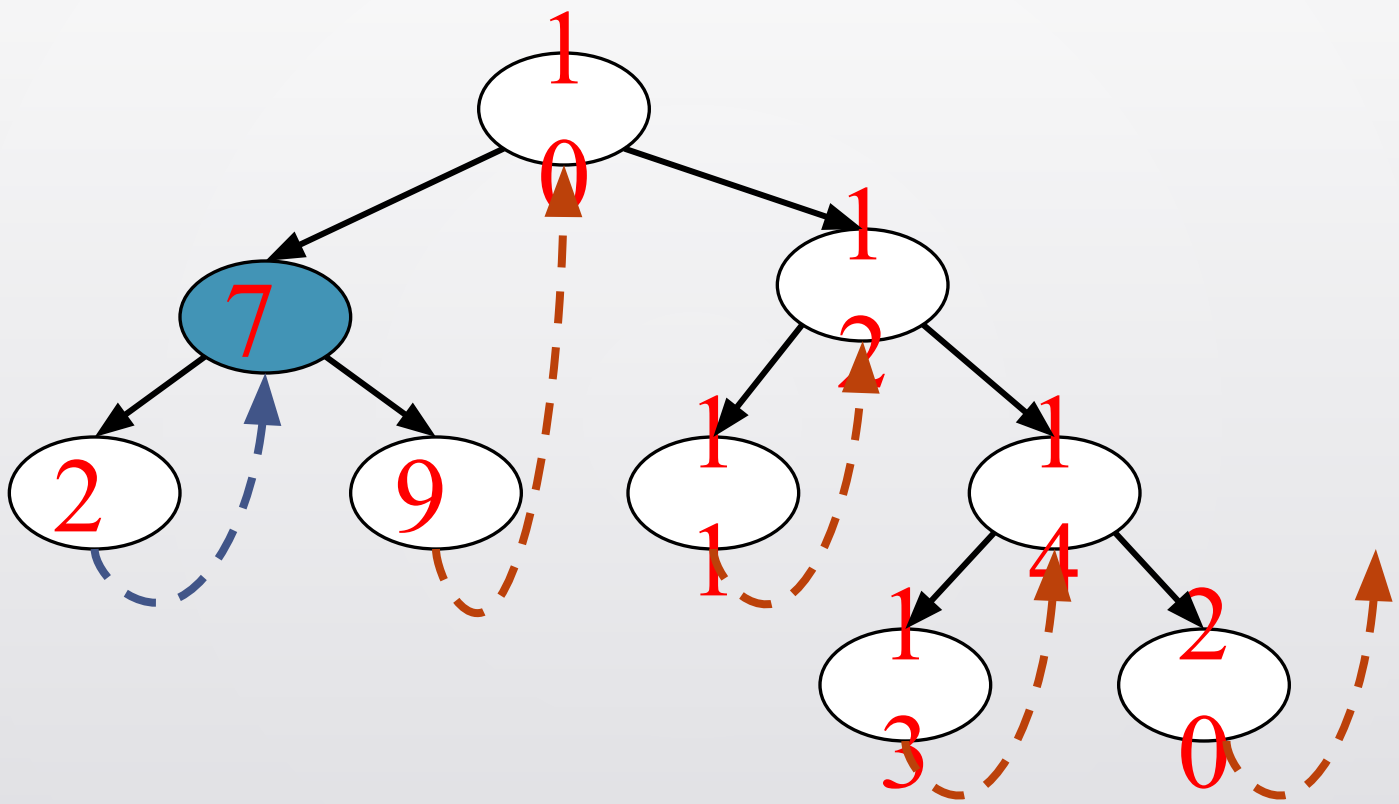
Threaded Binary Tree In-order Traversal



Output
2

Start at leftmost node, print it

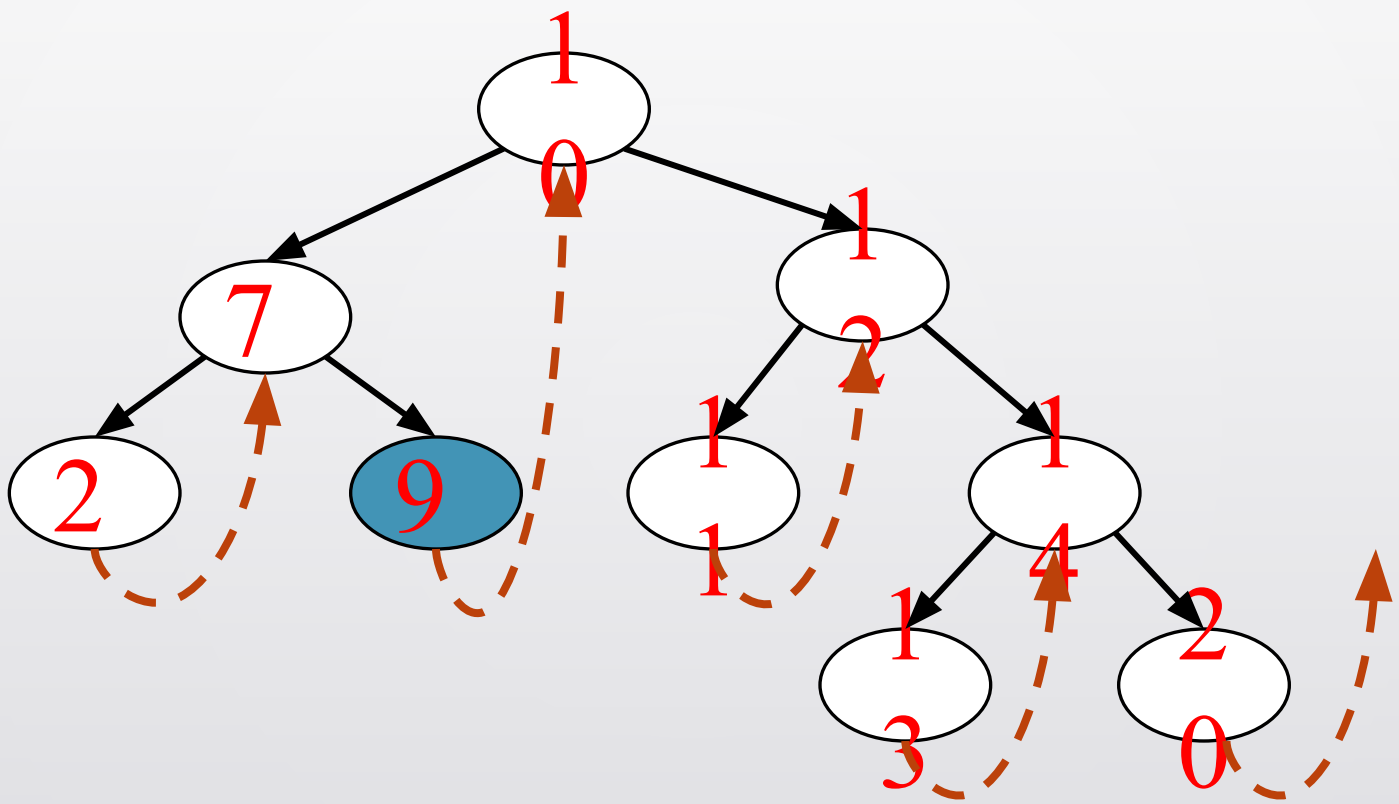
Threaded Binary Tree In-order Traversal



Output
2
7

Follow thread to right, print node

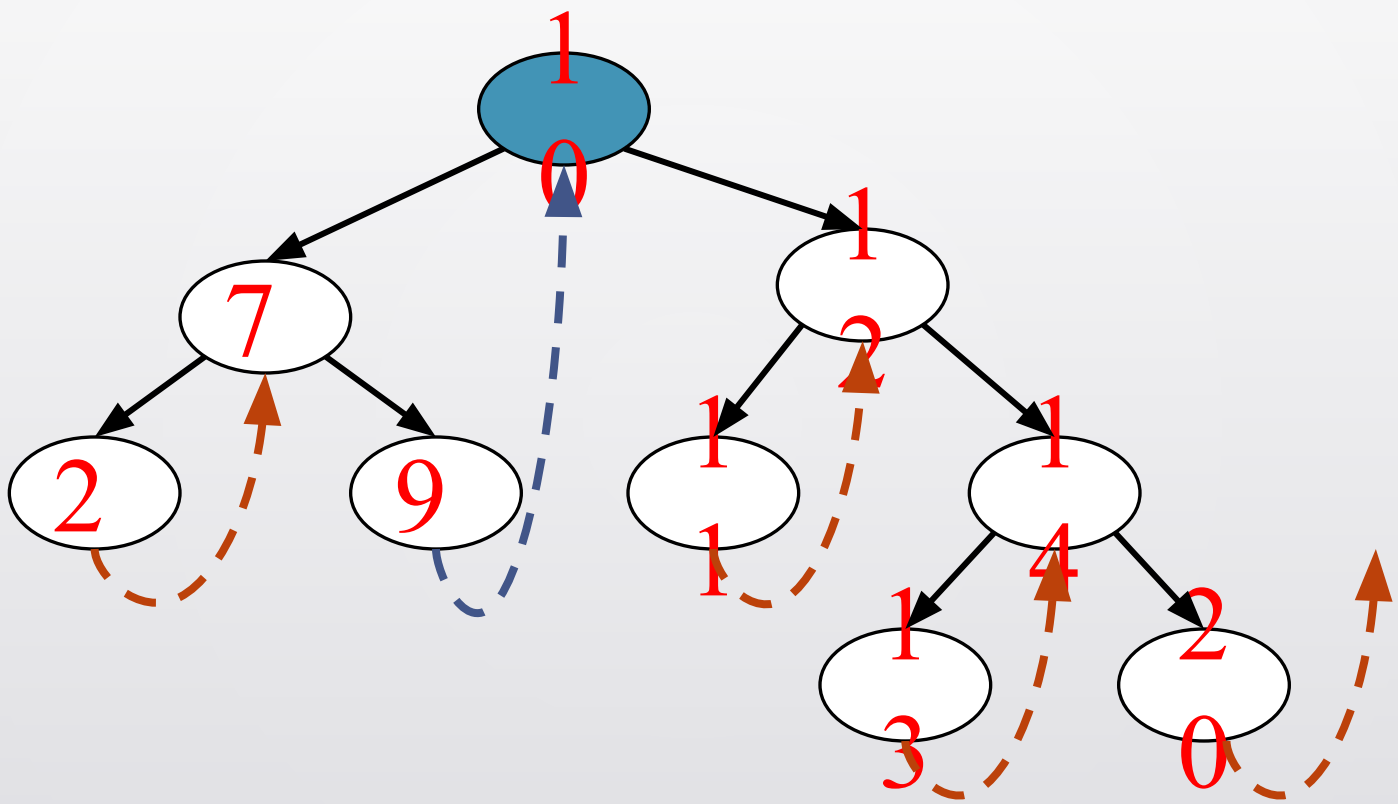
Threaded Binary Tree In-order Traversal



Output
2
7
9

Follow link to right, go to leftmost node and print

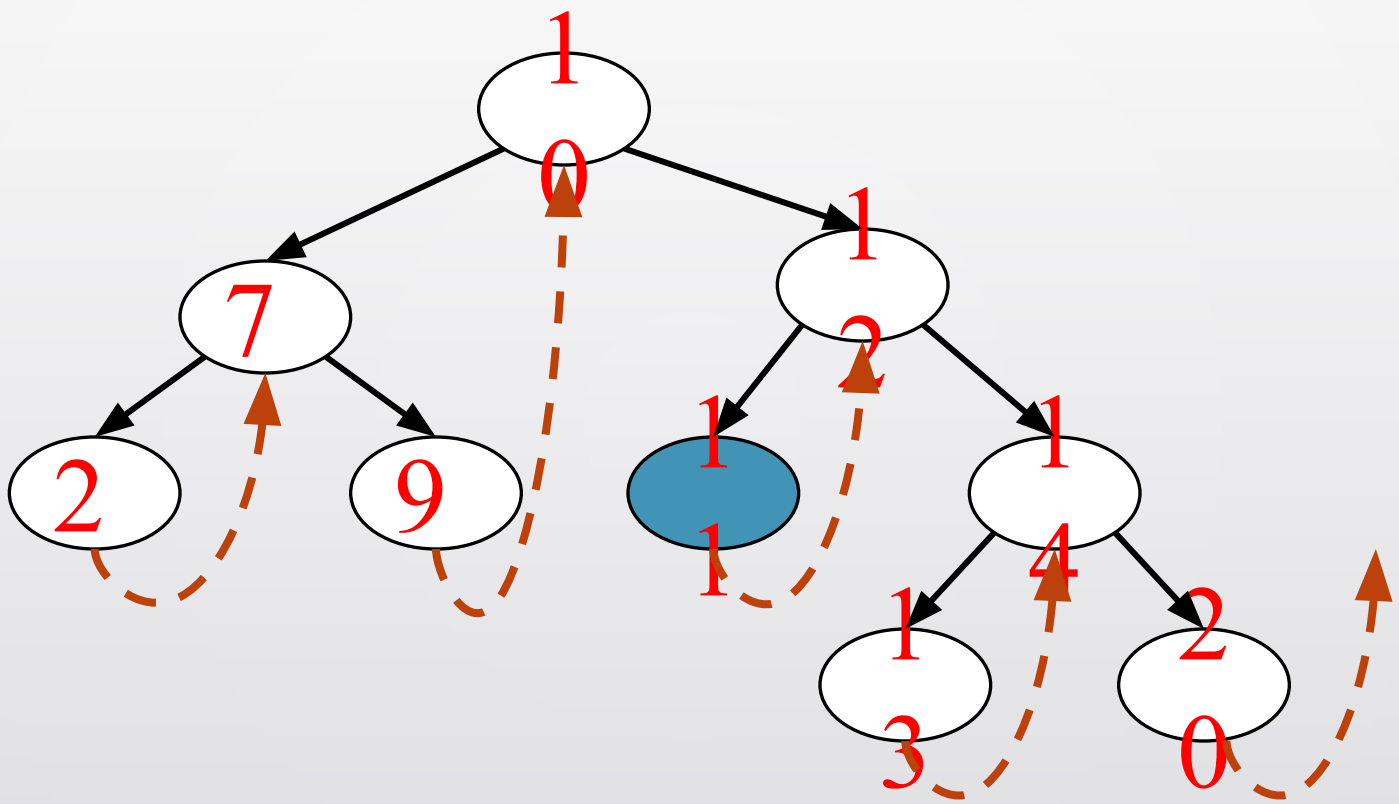
Threaded Binary Tree In-order Traversal



Output
2
7
9
10

Follow thread to right, print node

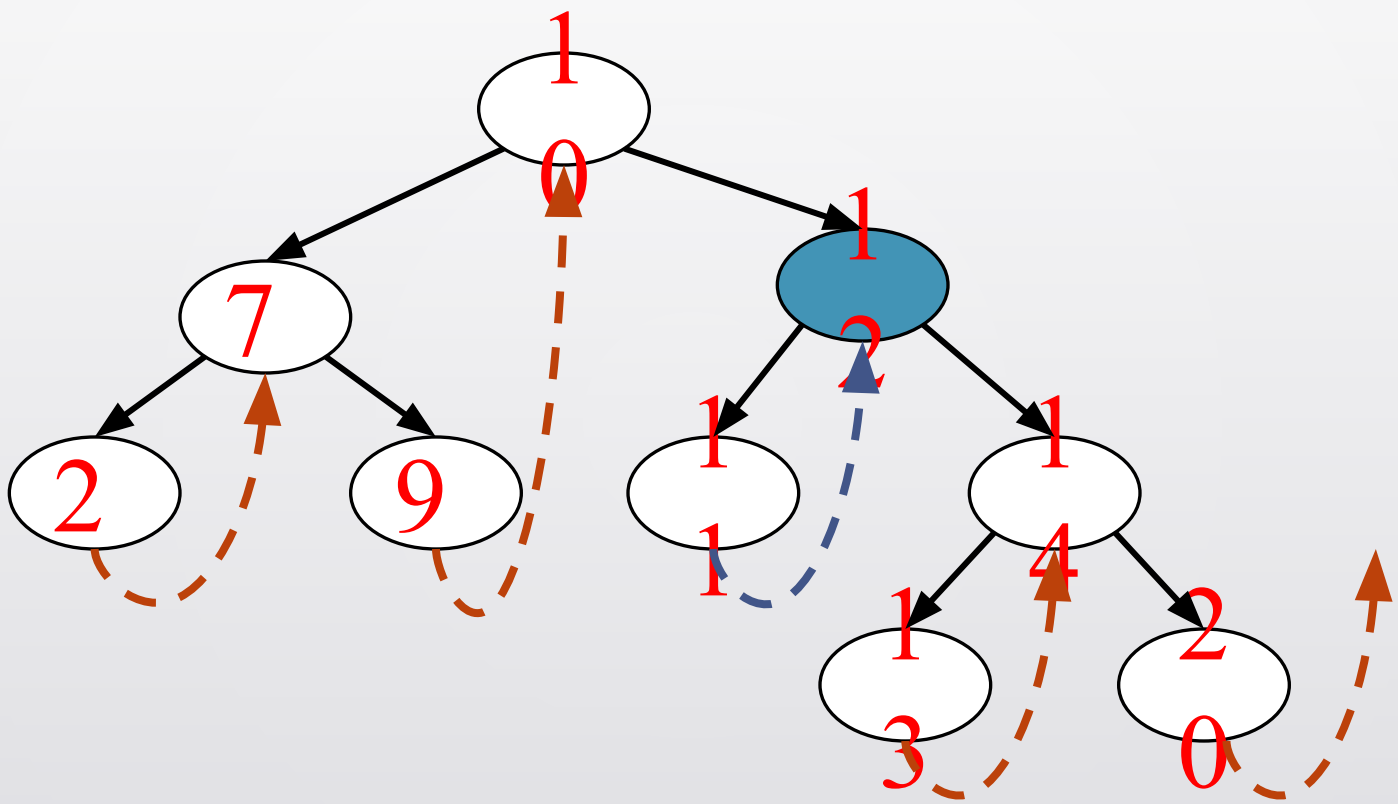
Threaded Binary Tree In-order Traversal



Output
2
7
9
10
11

Follow link to right, go to leftmost node and print

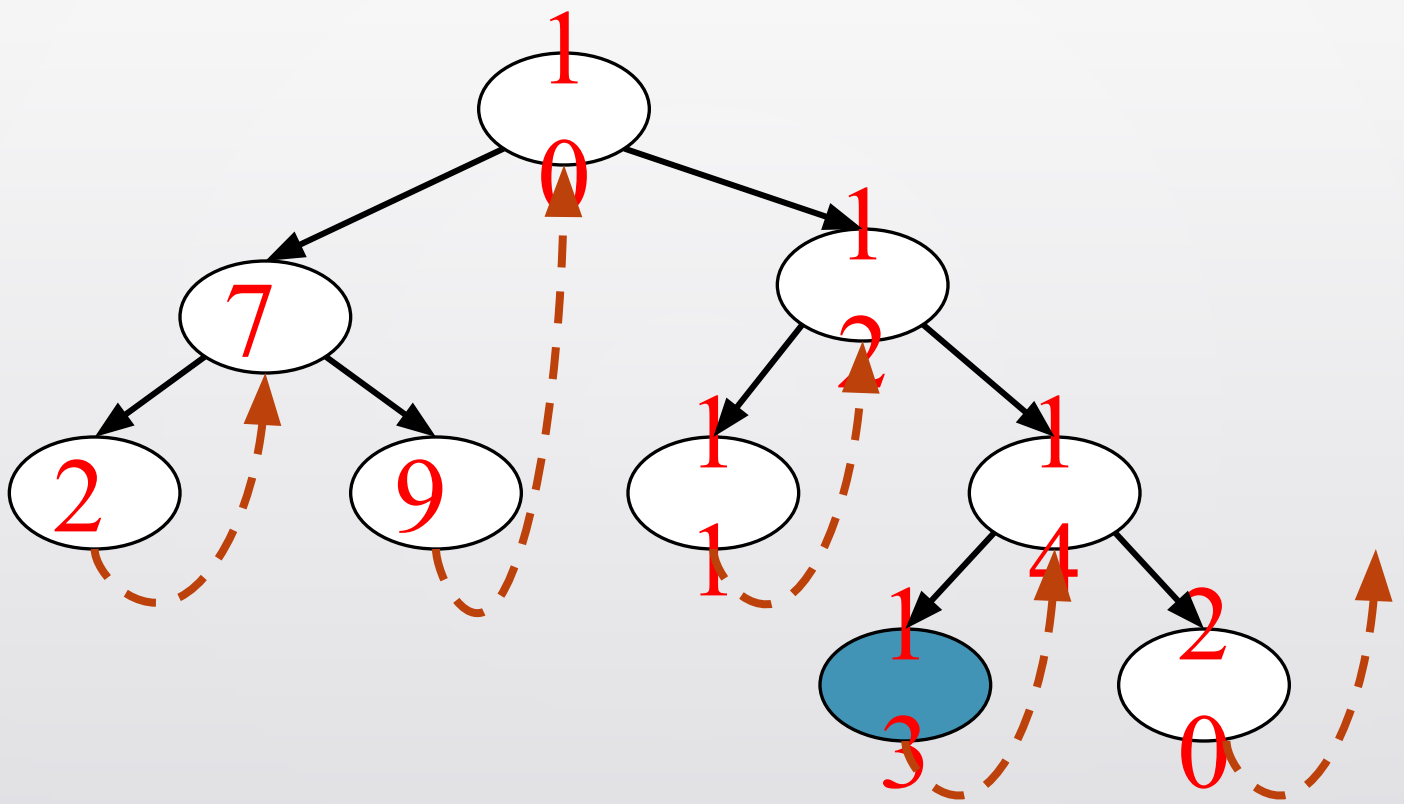
Threaded Binary Tree In-order Traversal



Output
2
7
9
10
11
12

Follow thread to right, print node

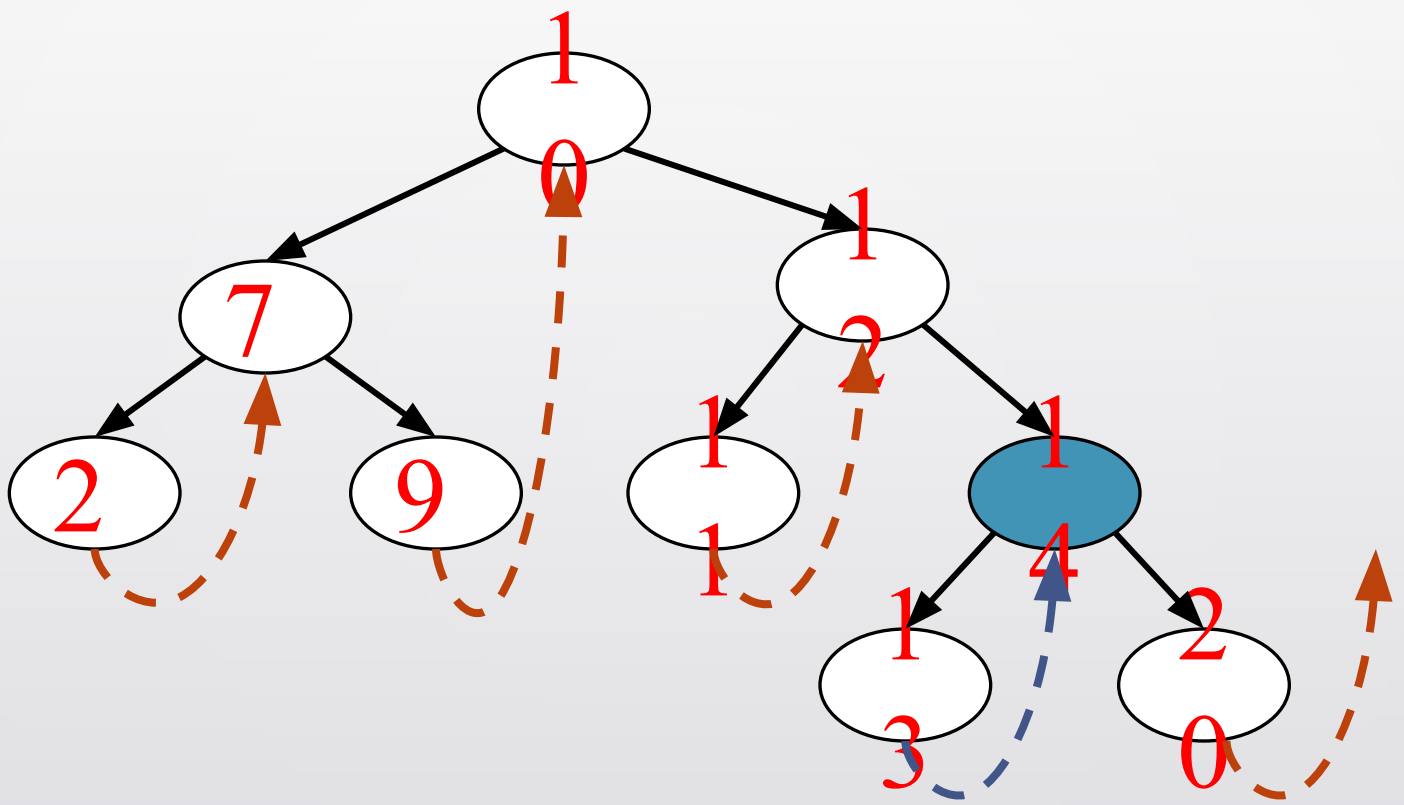
Threaded Binary Tree In-order Traversal



Output
2
7
9
10
11
12
13

Follow link to right, go to leftmost node and print

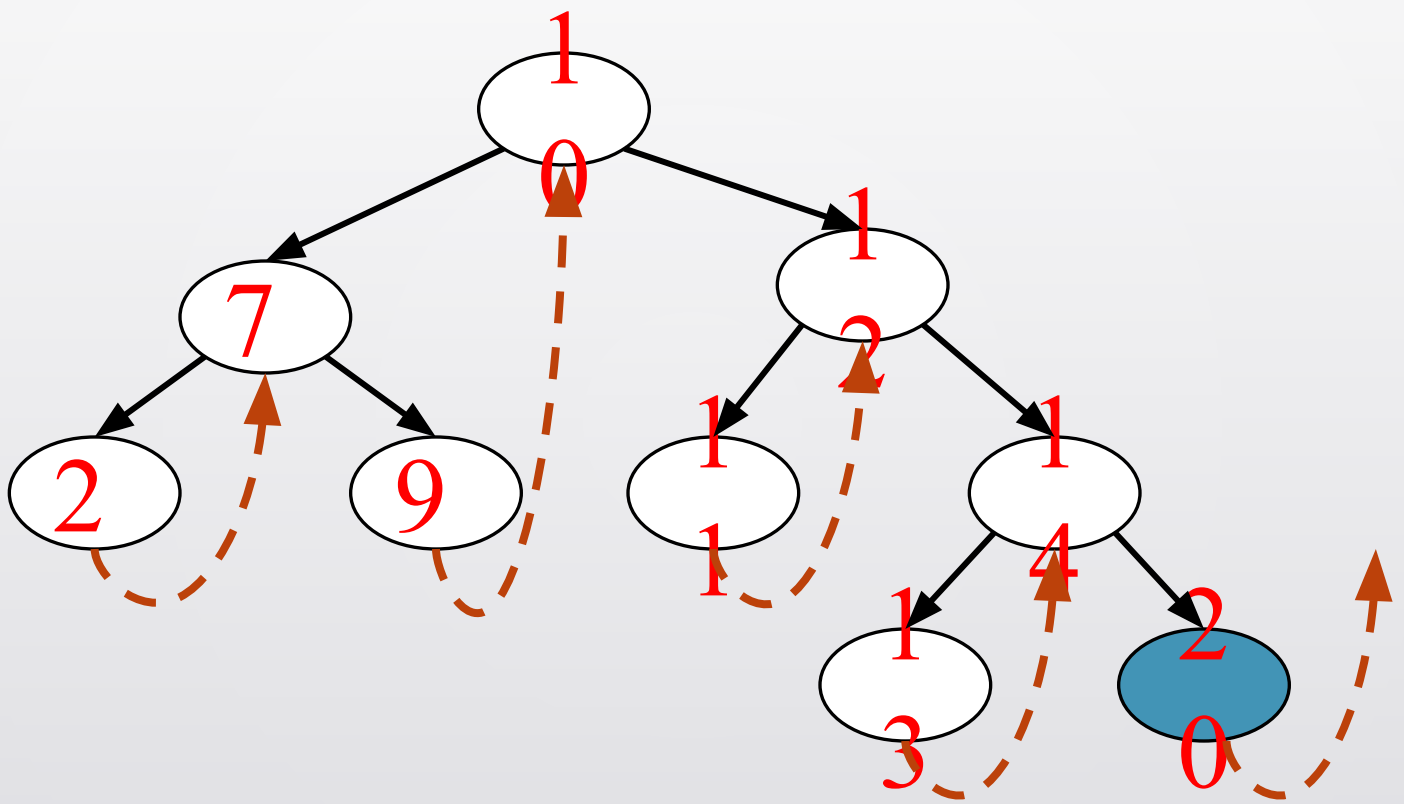
Threaded Binary Tree In-order Traversal



Output
2
7
9
10
11
12
13
14

Follow thread to right, print node

Threaded Binary Tree In-order Traversal



Output
2
7
9
10
11
12
13
14
20

Follow link to right, go to leftmost node and print

References

- Adam Drozdek, Data Structures and Algorithms in C++ (2nd Edition), 2001
- Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, "Fundamentals of Data Structures in C", Computer Science Press, 1992.
- https://www.geeksforgeeks.org/threaded-binary-tree/#:~:text=The%20idea%20of%20threaded%20binary,types%20of%20threaded%20binary%20trees.