Group No.: 19
Student IDs: 20595032, 20717304, 20700473, 20697931, 20610102, 20607620

Abstract

This report presents the Breadth-First Search algorithm for the traversal and sorting, in ascending order, of a binary tree containing 15 nodes using ARM assembly language. The BFS represents a level-order tree traversal scheme, in which all nodes level by level starting from the root are processed and the child is enqueued for further processing.

Node values of a binary tree: 10, 5, 30, 78, 2, 19, 11, 23, 48, 79, 1, 14, 9, 41, 31 First, these are traversed using BFS, and the level-order sequence is stored in an array. Then, the stored sequence is sorted using the Bubble Sort algorithm, which yields the final sorted order.

The ARM assembly code performs BFS traversal by using low-level instructions for node traversal, queue operations, and value storage. Further details of the data structures, algorithmic steps, and low-level implementation that allow this process are discussed in this report.

Introduction

Approach Implementation First of all, declare the main data structure: a queue, and the two pointers for a queue (`queueBegin` and `queueEnd`) to keep track of nodes waiting for being visited, and an array result of visited node values.

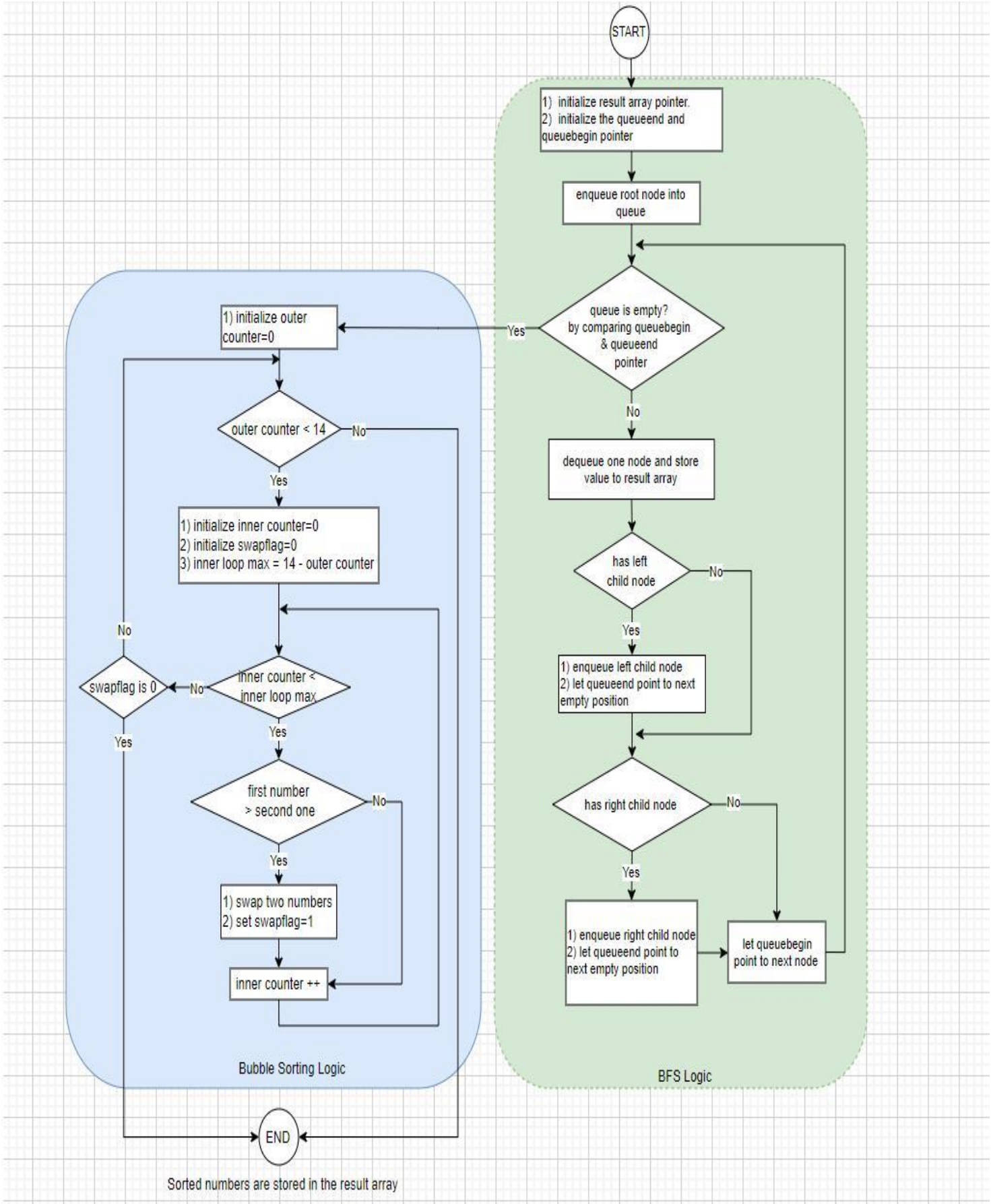
The traversal begins from the root node of the binary tree. The value of the root node is accessed and added to a queue, which acts as the starting point for BFS.

The BFS traversal can be done in an infinite loop, which doesn't end until the queue becomes empty. In every iteration, the front node in the queue is dequeued, whose value is pushed into the result array and increment the queueBegin pointing to the next node. If left and/or right children exist for the dequeued node, they are enqueued for subsequent processing. This indeed guarantees the visit of nodes in level order. The terminator can be checked by queueBegin and queueEnd pointers:.

After all nodes are processed, the BFS traversal is complete, with the node values stored in the result array. Following this, the Bubble Sort algorithm is used to sort this array in ascending order.

Conclusion

The Breadth-First Search algorithm in ARM assembly systematically traverses a binary tree by maintaining the enqueuing and dequeuing operations. Using a queue-based structure, the algorithm ensures that nodes are processed in order to enable the creation of a level-order sequence. This is then sorted into an ascending order of node values using the Bubble Sort method. The report shows how low-level ARM instructions can realize an efficient and structured way of doing binary tree traversal and sorting.


START
1) initialize result array pointer.
2) initialize the queueend and queuebegin pointer
enqueue root node into queue
queue is empty? by comparing queuebegin & queueend pointer
Yes
No
dequeue one node and store value to result array
has left child node
No
Yes
1) enqueue left child node
2) let queueend point to next empty position
has right child node
No
Yes
1) enqueue right child node
2) let queueend point to next node
let queuebegin point to next node
BFS Logic
1) initialize outer counter=0
outer counter < 14
No
Yes
1) initialize inner counter=0
2) initialize swapflag=0
3) inner loop max = 14 - outer counter
inner counter < inner loop max
No
Yes
swapflag is 0
No
Yes
first number > second one
No
Yes
1) swap two numbers
2) set swapflag=1
inner counter ++
Bubble Sorting Logic
END
Sorted numbers are stored in the result array