

Ques Report.

Adrian Jonsson Sjödin

Fall 2022

1 Introduction

2 Task

Implement two different FIFO queues. One utilizing arrays and one utilizing a linked list structure.

For the linked list queue, first implement a linked list queue structure that only has one property, the *head*, and new elements are simply added to the end of this list. What are the drawbacks to this implementation? What are the cost of removing the next element and adding a new element?

Change the linked list queue so that it holds a pointer to to the first element of the queue (the *head*), but also a pointer to the last element of the queue. This should allow one to add a new node at the end directly without first having to traverse the list to find the last node.

Now use the linked list queue to create a new iterator for the binary tree from last weeks assignment that traverses the tree breath first.

The array implementation of the queue should be dynamic and be able to increase in size, but also optionally shrink in size.

3 Method & Theory

4 Result

5 Discussion

Code

All the code can be found here: [GitHub](#)

Code Overview 1: Add to Binary Tree

Code Overview 2: Look up key in binary tree

Code Overview 3: `Next()` and `moveLeft(Node current)`