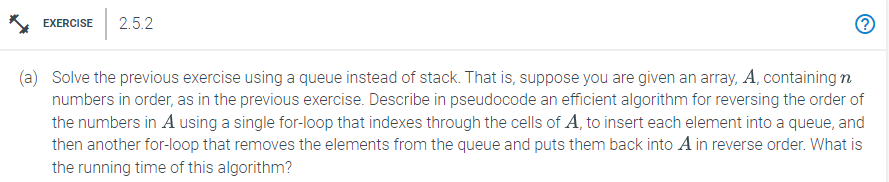
# Michael Chillemi

# 06/02/2023

# CS 590 - Algorithms

# M2.B1: Module 2 Basic Data Structures Reinforcement Exercises

Problem 2.5.2



Answer:

Algorithm reverseArrayUsingQueue(Input\_Array):

Input: Input\_Array

Output: Reversed\_Array

Queue = new Queue();

i = 0

For i <= length(A) -1,i++

Queue .enqueue(Input\_Array[i])

I = 0

For i <= length(A) -1, i++

A[i] = dequeue()

Return Reversed\_Array

The running time of this algorithm is . Each for loop has a run time of and runs for the length of the input array. For every enqueue and dequeue operation these functions have a run time. Since every for loop has a runtime of then this algorithm would be . Lastly we would drop the constant and the resulting runtime would be .

Problem 2.5.5

# 

Answer:

No, a binary tree cannot have a preorder traversal that is identical to a postorder traversal. Preorder traversal always leaves the root node as the first node it encounters. At the conclusion of postorder traversal, the root node is visited. Therefore, in an ordered tree with more than one node, the same preorder and postorder traversal cannot exist.

To answer the next question, yes it is possible to have the reverse postorder to be equal to the preorder. Consider this example, you have 3 nodes Node A, Node B, and Node C. Node A is the root node. Node B is a child of the Root Node and Node C is a Child of Node B. If you printed out the preorder of this binary tree it would be ABC. The print out for the postorder traversal is CBA. The reverse of the postorder is ABC which is the same as the postorder traversal.