Nicholas Skinner

CS 260

Design:

Design for this was fairly straightforward, take the previous assignment and implement two different search methods. One of them has to be recursive, and the other not so much. The non-recursive function borrowed a lot of itself from my replace method, I found that I could mostly copy and paste to get the end logic I needed. The recursive method would be a lot more difficult, as I need to keep track of what node I'm at, and not increment it at inappropriate times. From this I got an idea to call an initializer for the recursive method that will declare values to be used by it, but should not be modified by it.

Testing:

As they both take the same approach of checking value by value, I'm looking for both of these methods to come up with the same number of results to get to their end position. Something that I came into while testing, was that I neglected to setup input validation. I could re-design, and likely will before I submit my project to not include user input for this segment.

Test case 1 - Correct input

Gives desired output, comparisons are the same.

Test case 2 – invalid non-numeric input

Returns an error, as expected due to no input validation

Test case 3 – Invalid numeric input

Returns error, as value entered may not be within List's defined range

Reflection:

This was an interesting assignment, it took me a bit of research to get to my recursive method, but once I got there it wasn't so difficult to implement the non-recursive one. It was interesting to learn how much they have in common in this instance, and yet how different they are to implement in a search function. I hadn't really had the chance to work with recursion much before this assignment, but now I feel a much stronger grasp on what it is, and how to create a loop based on it. Nodes, and the general on paper 'structure' of these is really helping me get the method to programming them, but there are still snags that I run into from time to time that I really don't understand what I'm doing, but It seems that those times are the best to learn some new trick or approach to the problem.