Cesar Santiago

COP 3530 Data Structures & Algorithms

Singly Linked List:

Display function utilizes a O(n) since it has to go through the entirety of the list only once to display the data points.

Insert function utilizes an O(1) because the program uses a tail pointer which means that to insert another node you only need one command, and there is also no loops.

Sorting became more complicated, I used a weird version of mergesort that allowed for my sorting to be recursive. After the many while and for loops I came to consider this a O(n\*log(n)) algorithm because after the recursion, there was no nested loops. The added while and for loops on the assisting functions account for the ‘n’ and the recursiveness for the ‘log(n)’.

Stacks:

PrintList function served the same purpose as the display function on the linked list algorithm and therefore accounts as an O(n) algorithm.

Push is a function that adds a node at the beginning of the list essentially making it a stack.

Because it uses the head pointer to add new nodes it has an O(1).

Pop takes out the first node on the stack and returns the value to the user. Since it uses the head pointer it has an O(1).

GetUserInput is a helper function that runs only once.

Menu is a helper function that runs only once.

The main function itself uses an unrestricted while loop functionally making this a sequential algorithm, therefore it runs O(n).