Question 2

Pushing elements onto the stack:

The amount of time needed to push a single element is constant time. Referring to the provided Stack.cpp file, to do so, we would need to do the following:

- Create a new node
- Set the new node's data to the data (element) that we are wanting to push
- Set the next pointer of the newly created node to null because there won't be any nodes to link to after the element that we are pushing onto the stack
- Check if the current stack is empty or not
- If it is empty, we are going to assign the newly "pushed" element as the tail of the SHSL
- If it is not empty, we are going to link the newly created node to the last node and then update the 'top' pointer to that element

All the steps above require linear time to achieve. As a result, the max total time needed to push a single element is O(n).

Pushing n elements onto the stack requires us to perform this push method n times. Therefore, this gives us a total time of $O(n^2)$.

Popping elements off the stack:

The amount of time needed to pop a single element is constant time. Referring to the provided Stack.cpp file, to do so, we would need to do the following:

- Check if the stack is empty → if it is, we are just going to return it
- Check if there is currently one element in the stack → if there is, we are going to delete that node, and set everything to null
- If the current stack is not empty nor does it contain one element, delete the node and update top pointer to the next node and set pointers to null (popping the tail of the linked list)

All of the steps above linear constant time to achieve. As a result, the max total time needed to pop a single element is O(n).

Popping n elements onto the stack requires us to perform this pop method n times. Therefore, this gives us a total time of $O(n^2)$.