Q2: Time Efficiency Analysis

**Referring to the Stack ADT class you have implemented in Question 1, analyze the total time required to push n elements to the Stack and express it using the Big O notation.**

The Stack ADT class implemented in Question 1 is a SHSL linked-list with two class invariants: Last-In-First-Out (LIFO) and location of the Stack’s “top” at the back (Last node of the linked list). Since we are working with a SHSL list (Single reference to the head node), we must traverse through the entire list of *n* elements before adding the next node to the Stack. Therefore, we can say that the time complexity of push() is O(*n*) as it is dependent on the size of the linked list.

Pushing n elements on the linked list… (*m* is the size of the linked list so far)

(m+1) + (m+2) + … + (m+n-1) + (m+n)

S(n) = m\* summation (n, k=1)k…

= n(n+1)/2 (2)

= n^2/2 + n/2

= O(n^2)????

**Next, analyze the total time required to pop those n elements from the Stack and express it using the Big O notation.**

Similarly, to question 1, the pop() operation requires us to traverse through the entire list and remove the “top” element of the Stack. Therefore, we can also say that the time complexity of pop() is O(n) as it is dependent on the size of the linked list.

To pop *n* elements from a Stack of size *m*… we need to

^^ Check above first

**To Remove**

A detailed analysis is expected, i.e., if you present a final answer only, you will be unhappy with your grade. In other words, "show your work".

Type your analysis and save it in a file called Analysis.pdf and submit this file to CourSys.