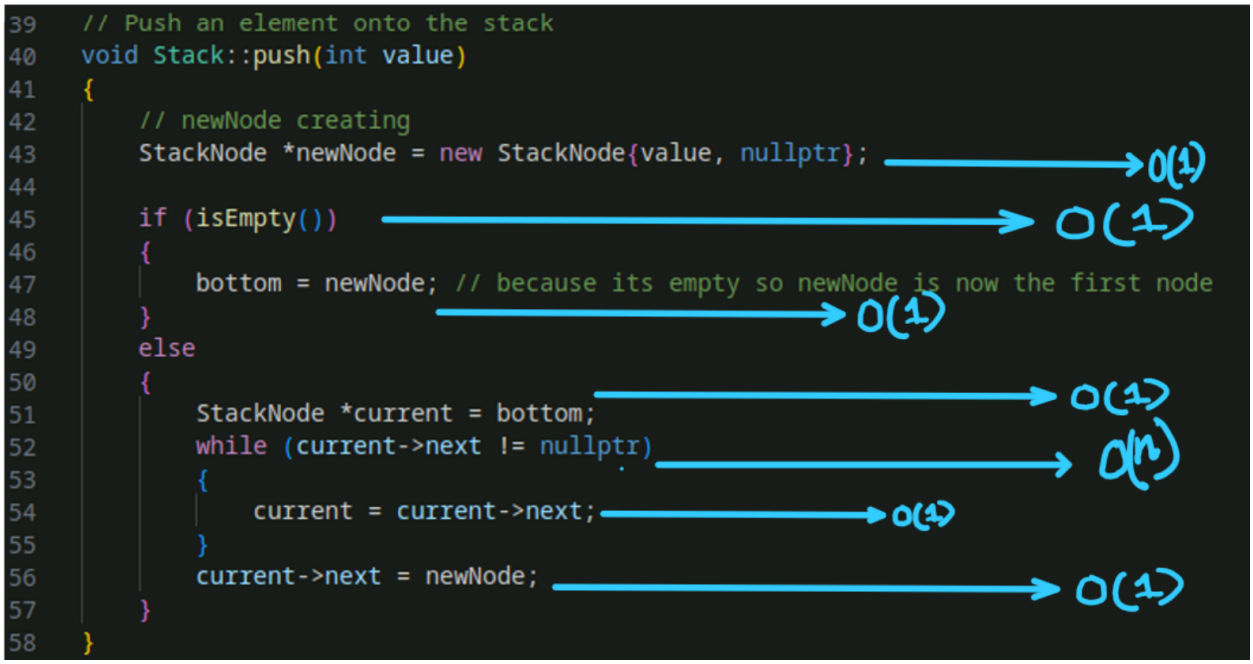
Author: Fardin Abdulla

Date: 16/02/2024

Time Efficiency Analysis

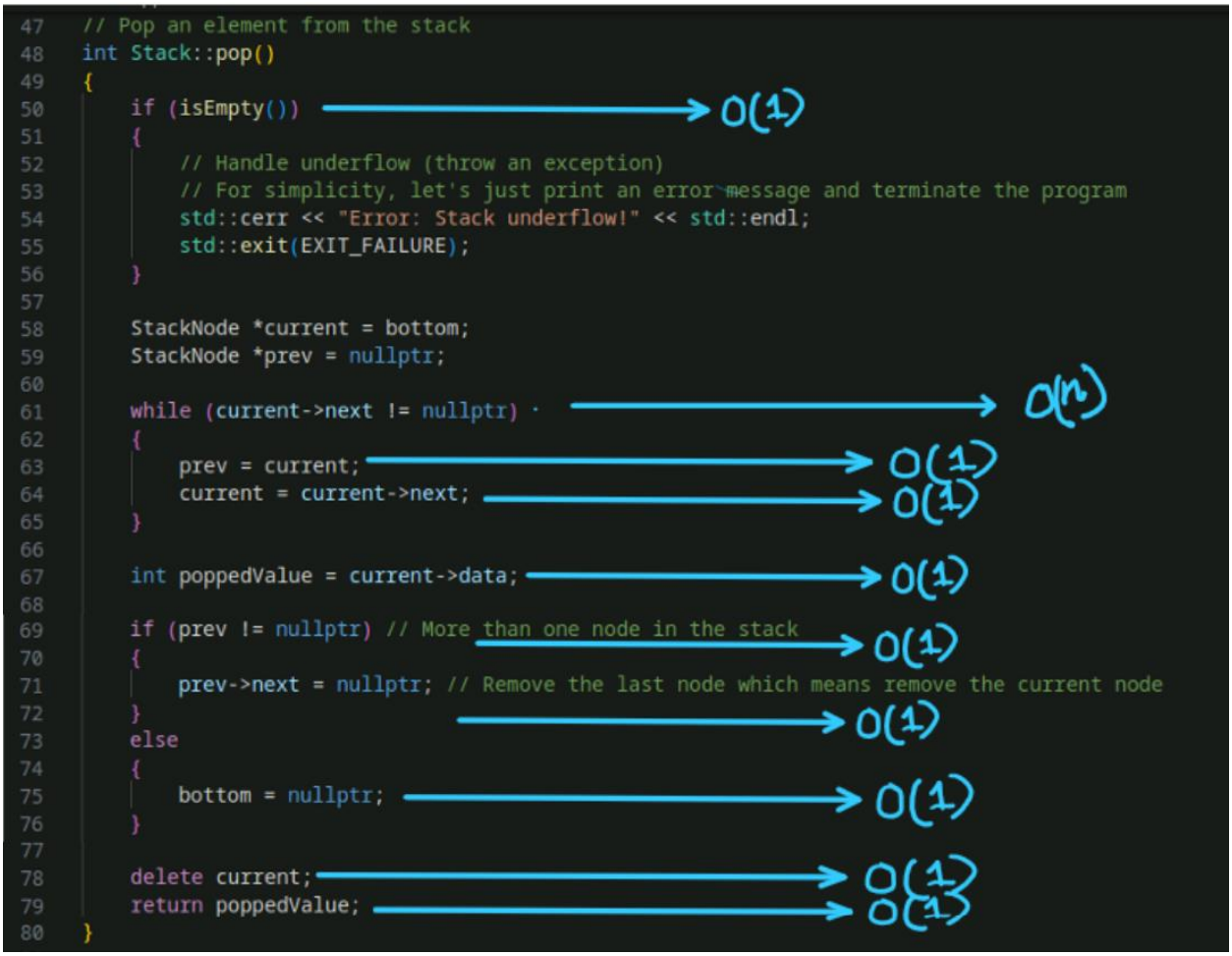
1.push():



Overall time efficiency of Algorithm pop(): max[O(1), O(1), O(1), O(n), O(1)] = O(n)

The push operation involves adding a new element to the back of the singly linked list. Since the top of the stack is located at the back of the list, we need to traverse the entire list to find the last node and then add a new node after it. But, as we are analyzing the total time required to push n elements to the Stack, it would be = [O(n) * O(n)] = $O(n^2)$

2. pop():



Overall time efficiency of Algorithm pop(): max[O(1), O(n), O(1), O(1), O(1), O(1), O(1)] = O(n)

The pop operation involves removing the last element from the back of the SHSL. So, the time complexity is O(n). But, as we are analyzing the total time required to pop those n elements from the Stack, it would be = [O(n) * O(n)] = $O(n^2)$