**ASSIGNMENT 2 REPORT**

By: Khadija Swailem 202301222

***Assumptions:***

I assumed stack is of unlimited size

***Design approach:***

**Attempt 1:**

Since the assignment stated that we use two stacks to implement a queue I drew the two stacks and drew the queue and used one stack for all the enqueue operations of the queue using push method, and then used the other stack for when I needed to dequeue so I popped all the elements from stack 1 and then pushed them in stack 2 , so now the order is reversed which will help implement the queue concept FIFO (first in first out) and then I popped the top of the stack as it uses LIFO (last in first out) and then popped all the elements from stack 2 and pushed them in stack 1 to reverse their order again back to original before dequeuing but without item we dequeued. I think this is the most optimum approach within the limits set as we move the elements once to stack 2 then pop the top to dequeue. And as for the enqueue operation I used push() function to enqueue it into the queue.

**Attempt 2:**

In the second attempt, since I had more freedom, I decided to implement a function called returntop() which uses getitem() function from the linked list class to access the item at index 0 which would be what we want to dequeue and then I passed the result of the returntop() to the pop function inside the dequeue function so it can pop it . And for the enqueue function I used the push method only as done in attempt 1. I think this is an optimum approach because since I can directly access the element to be popped the time complexity is constant.

***Big-O analysis:***

**Attempt 1:**

Time complexity of Enqueue operation: O(1).Since we just use push method which uses append and both have time complexity O(1).

Time complexity of Dequeue operation: O(n).Since we pop n times to get the elements out of stack 1 and then push n times to put them in stack 2 then we pop once to remove the top element(element to be dequeued) and then we pop n-1 times and push n-1 to return elements to stack 1 again to get their original order but after dequeuing.

**Attempt 2:**

Time complexity of Enqueue operation: O(1). Since we just use push method which uses append and both have time complexity O(1).

Time complexity of Dequeue operation: O(1).Since we call the pop function for one item and it uses returntop() which also has complexity O(1) and since we’re getting the item at index 0 it iterates only once so best case of iterating using for loop as we don’t iterate n times to reach index n so also O(1).