CCC2113 Data Structures and Algorithms Analysis **Name**:

Lab 8: List & Stack (6/6/2024) **Group**: A

1. Provide practical application examples for:

    a. Singly linked list: Implementing stacks and queues, Navigation in web browsers

    b. Doubly linked list: Implementing music players, Implementing text editors

2. What is the worst case time complexity for searching an element in a singly linked list. Justify your answer.

- In a singly linked list, elements are not indexed, and each node points to the next node in the sequence. To search for an element, you must start from the head node and traverse the list one node at a time until you find the target element or reach the end of the list.
- In the worst-case scenario, the element you are searching for is either at the very end of the list or not present at all. This means you would have to check each of the n elements in the list once, resulting in a time complexity of O(n).

3. Write the algorithm steps for push operation in a stack

```java
class Stack {
    int[] stackArray;
    int top;
    int maxSize;

    // Constructor to initialize the stack
    Stack(int size) {
        stackArray = new int[size];
        maxSize = size;
        top = -1; // stack is initially empty
    }

    // Push operation
    void push(int value) {
        // Step 1: Check for stack overflow
        if (top == maxSize - 1) {
            System.out.println("Stack Overflow");
            return;
        }

        // Step 2: Increment the top pointer
        top++;

        // Step 3: Add the new element
        stackArray[top] = value;
    }

```