Data Structures

Quiz - 2

Question:

(a) Implement the following method:

```
public static <E> boolean search(Stack<E> queue, E element)
```

Ensure that when your method returns, the stack remains unchanged from its original state.

(b) If the enqueue and dequeue operations have an asymptotic complexity of $O(1)$ for a particular implementation, what are the worst-case and best-case asymptotic computational complexities of your method?

Solution:

There are 2 possible solutions! (For both, Asymptotic complexity is O(n))

```java
public static <E> boolean search(Stack<E> queue, E element) {
    boolean isFound = false;
    Queue<E> searchQueue = new LinkedQueue<E>();
    Stack<E> reversed = new LinkedStack<E>();

    while(queue.size() != 0) {
        searchQueue.enqueue(queue.pop());
    }

    while(searchQueue.size() != 0) {
        E searchElement = searchQueue.dequeue();
        if (searchElement.equals(element)) {
            isFound = true;
        }
        reversed.push(searchElement);
    }

    while(reversed.size() != 0) {
        queue.push(reversed.pop());
    }

    return isFound;
}
```

```java
public static <E> boolean search(Stack<E> queue, E element) {
    boolean isFound = false;
    Stack<E> reversed = new LinkedStack<E>();

    while(queue.size() != 0) {
        E searchElement = queue.pop();
        if (searchElement.equals(element)) {
            isFound = true;
        }
        reversed.push(searchElement);
```

```
    }

    while(reversed.size() != 0) {
        queue.push(reversed.pop());
    }

    return isFound;
}
```