Kayikunmi Babatunde - Akinmagbe
COSC - 211 - 02
HW3

1.) $4 < \lg n < 2^{\lg n} < 100n < \sqrt{n} < 2^n < n!$

2.) $f(n) = 2n^2 + 6n - 9$ $\quad$ $f(n) \in O(n^2)$

Choose : $c=3$, $n_0=2$

want to show : $\forall n \geq 2$

$0 \leq f(n) \leq c \cdot g(n)$

$0 \leq 2n^2 + 6n - 9 \leq 3n^2$

$0 \leq 2n^2 + 6n - 9$ is true, so ...

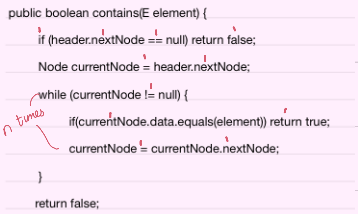
$2n^2 + 6n - 9 \leq 3n^2$

$2n^2 + 6n - 9 \leq 2n^2 + n^2$

$6n - 9 \leq n^2$ is true

$\therefore f(n) \in O(n^2) \quad \forall n \geq 2$

3.)



```
public boolean contains(E element) {
        if (header.nextNode == null) return false;
        Node currentNode = header.nextNode;
        while (currentNode != null) {
                if(currentNode.data.equals(element)) return true;
                currentNode = currentNode.nextNode;
        }
        return false;
```

Worst case: $O(n)$ if currentNode is not null, the loop will repeat n times till currentNode is null.

Best case: $O(1)$ when currentNode is the first node, and is null, So, the loop will not repeat.

4a)

	Stack version 1	Stack Version 2
push()	$O(1)$: add to the end of array. The function will not need a loop	$O(n)$: we need a loop that will repeat its operations n times, to add to the second stack
pop()	$O(1)$: just remove the item last used. The function will not be repeated.	$O(1)$: just remove the item last used. The function will not be repeated.

4b)

	Stack version 1	Stack Version 2
add ()	O(n): we need 2 loops to push into the second array, add the item then pop back into main array so, O(n) + O(n) = O(n). (Repeats n times)	O(n): we need 2 loops to push into the second array, add the element into an array with more space and then use the other loop to add the removed elements
remove ()	O(1): just remove the item last used. The function will not be repeated.	O(1): just remove the item last used. The function will not be repeated.