Kyla Wilson
J00813814
Assignment 5
CSC 323

```
Left: 0, Right: 11, Mid: 5
Left: 0, Right: 4, Mid: 2
Left: 3, Right: 4, Mid: 3
Left: 4, Right: 4, Mid: 4
********************
Left: 0, Right: 11, Mid: 5
Left: 6, Right: 11, Mid: 8
Left: 6, Right: 7, Mid: 6
Left: 7, Right: 7, Mid: 7

Array:
1 1 3 4 5 5 5 5 7 9 9 14
Target value: 5

First occurance at index: 4
Last occurance at index: 7
Total occurance count: 4
Program ended with exit code: 0
```

```
Left: 0, Right: 11, Mid: 5
Left: 6, Right: 11, Mid: 8
Left: 6, Right: 7, Mid: 6
Left: 7, Right: 7, Mid: 7
********************
Left: 0, Right: 11, Mid: 5
Left: 6, Right: 11, Mid: 8
Left: 6, Right: 7, Mid: 6
Left: 7, Right: 7, Mid: 7

Array:
1 1 3 4 5 5 5 5 7 9 9 14
Target value: 6
Number does not appear in array
Program ended with exit code: 0
```

```
Left: 0, Right: 11, Mid: 5
Left: 6, Right: 11, Mid: 8
********************
Left: 0, Right: 11, Mid: 5
Left: 6, Right: 11, Mid: 8

Array:
1 1 3 4 5 5 5 5 7 9 9 14
Target value: 7

First occurance at index: 8
Last occurance at index: 8
Total occurance count: 1
Program ended with exit code: 0
```

0 1 2 3 4 5 6 7 8 9 10 11

Array: {1, 1, 3, 4, 5, 5, 5, 5, 7, 9, 9, 4}

(ii)

Iteration	left	Right	Mid
0	0	11	5
1	0	4	2
2	3	4	3
3	4	4	4

First Index = 4

Iteration	left	right	Mid
0	0	11	5
1	6	11	8
2	6	7	6
3	7	7	7

Last Index = 7

(iv)

Iteration	left	Right	Mid
0	0	11	5
1	0	4	2
2			

First Index = 2

Iteration	left	Right	Mid
0	0	11	5
1	0	4	2

Last Index = 2

v)

Iteration	Left	Right	Mid
0	0	11	5
1	6	11	8
2	6	7	6
3	7	7	7

First index is still -1

Iteration	Left	Right	Mid
0	0	11	5
1	6	11	8
2	6	7	6
3	7	7	7

Last index is -1 ∴ 6 does not appear in the array.

ii) Since the binary search algorithm is O(logn). My solution will still be O(logn) even though I do it twice.

i) The invarient changes when the middle element is less than or greater than the target. In both algorithms. Once left is greater than right, the loop terminates.