First Occurrence of a Number#

When given an array, find the first occurrence of a given number in that array and return the index of that number.

The following illustration explains how to approach this problem.

```
O 1 2 3 4 5 6
2 3 4 1 7 8 3

First index of 3
O 1 2 3 4 5 6
2 3 4 1 7 8 3

Output: 1
```

The following code explains how to find the first occurrence of a number in an array.

Implementing the Code

Experiment with the code by changing the values of array and num to see how it works!

1 class ArrayClass {

```
private static int firstOccurrence(int[] a, int n, int currentIndex) {
        if (a.length == currentIndex) {
          return -1;
        else if (a[currentIndex] == n) {
        return currentIndex;
 9
        else {
10
            return firstOccurrence(a, n, currentIndex+1);
11
12
13
      }
14
      public static void main(String[] args) {
15
        System.out.print("{");
16
17
        int[] array = \{2,3,4,1,7,8,3\};
18
        for (int i = 0; i < array.length; i++) {</pre>
19
          System.out.print(array[i] + " ");
20
21
        System.out.println("}");
22
23
        int num = 3;
24
25
        int result = firstOccurrence(array, num, 0);
26
        System.out.println("The first occurrence of the number " + num + " is at index: " + result);
27
28
Run
                                                                                                    Reset
```

The driver code is found from line 16 to line 27.

Understanding the Code

Driver Method#

The code snippet above can be broken down into two parts. The recursive method and the main where the

• From **line 18** to **line 24** we have an **array** equal to of 7 and the number **num** whose first occurrence is to be found respectively.

method is called.

• On line 26 the method firstOccurence is defined and takes in 3 arguments i.e., array, num and 0 (i.e.

is not found, the method returns -1.

- the first index). When the method is called, it returns the index of the first occurrence num.
- Recursive Method#

 In the recursive method, we define the base case and the recursive case.

The method terminates when one of the following conditions is met:

Base Case

• The first base case is defined from **line 4** to **line 6**. If the whole array has been traversed and the element

Find first occurrence of 3

Stack is empty

index of that number is returned.

Recursive Case#

• The second base case is defined from line 7 to line 9. If the first occurrence of the number is found, the

The recursive case is called on line 11.
 The method takes in three arguments. The first argument is the array a. The second is the number n

whose first occurrence is to be located. The third and argument is the currentIndex which is incremented in successive recursive calls.

• Initially, the value of the currentIndex is 0. Each time the recursive method is invoked, the value at the

result of the comparison is true, the method terminates and returns the index of the n.If the result of the comparison is false, the method **increments the currentIndex** in order to compare n with the value stored at the next index of the array. It then keeps comparing until it reaches the base case.

currentIndex of the array is compared with the number n whose first occurrence is to be located. If the

Understanding Through a Stack#

The following illustration explains this concept:

5

3

1

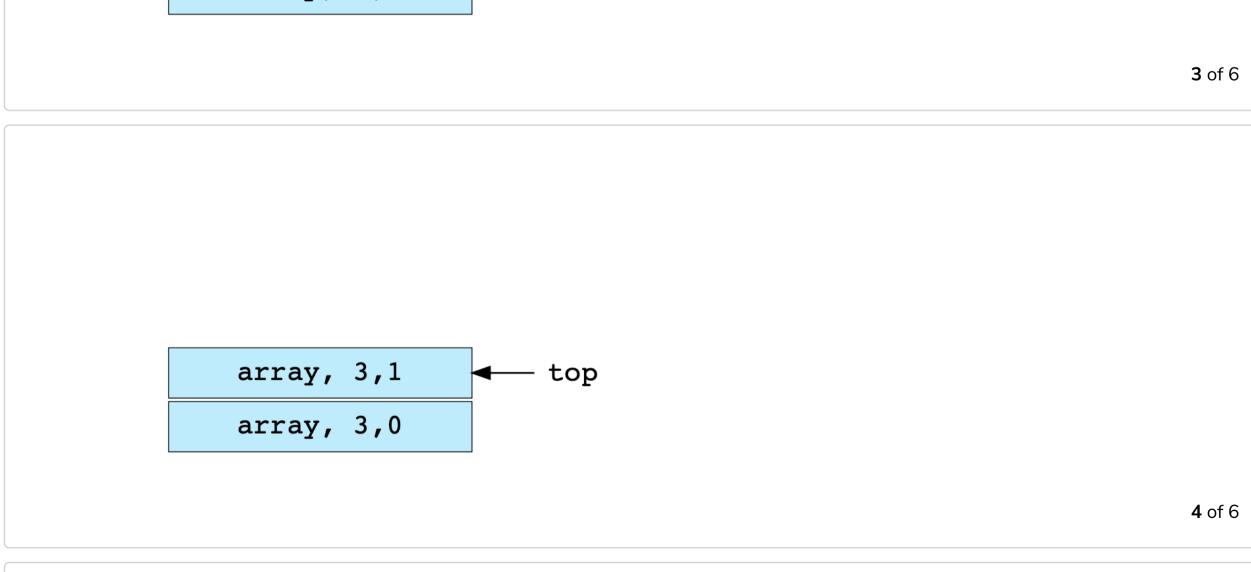
```
0 1 2 3 4 5 6
2 3 5 7 9 1 3

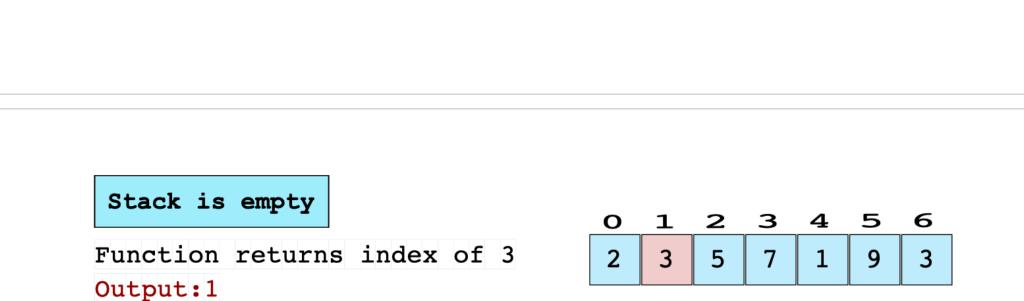
if array[0]==3 => False

2 of 6

0 1 2 3 4 5 6
2 3 5 7 9 1 3

if array[1]==3 => True
//base case
stack starts popping at this point
```





array, 3,0

5 of 6

6 of 6