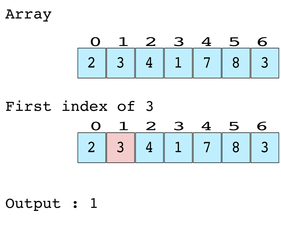
First Occurrence of a Number

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.



Implementing the Code

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

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

```
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;
    }
    else {
        return firstOccurrence(a, n, currentIndex+1);
    }
  }

  public static void main(String[] args) {
    System.out.print("{");

    int[] array = {2,3,4,1,7,8,3};
    for (int i = 0; i < array.length; i++) {
      System.out.print(array[i] + " ");
    }
    System.out.println("}");

    int num = 3;

    int result = firstOccurrence(array, num, 0);
    System.out.println("The first occurrence of the number " + num + " is at index: " + result);
  }
```

Run Save Reset

Understanding the Code

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

Driver Method

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

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

Base Case

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

- The first base case is defined from **line 4** to **line 6**. If the whole array has been traversed and the element is not found,the method returns -1.
- The second base case is defined from **line 7** to **line 9**. If the first occurrence of the number is found, the index of that number is returned.

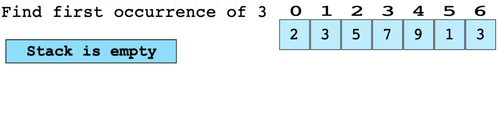
Recursive Case

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

Understanding Through a Stack

The following illustration explains this concept:



1 of 6

In the next lesson, we'll determine the number of occurrences of a particular number in a given array.