



First Index (recursive)

Problem Description: Given an array of length N and an integer x, you need to find and return the first index of integer x present in the array. Return -1 if it is not present in the array.

Sample Input:

5	(Size of array)
1 4 -1 3 4	(The array elements)
4	(x)

Sample Output:

1

How to approach?

Doing this iteratively is trivial. All we need to do is to start a loop from the 0th index to the end of the array. If we encounter 'x', then we return the index we're currently on. If the whole loop is finished and we didn't return any value, that means 'x' is not present on the array, so we return -1.

We can do the same process recursively. Let's create a function that returns an integer and takes in the following parameters:

1. The array itself
2. x
3. A start index (initially 0)

Now, we will check whether the value element at index 'start' is equal to 'x'. If yes, then we straight-away return 'start', otherwise we recurse on our function passing the following arguments: the array, x and start + 1

What will be the base case? The most trivial input would be an array of size 0. In that case, we won't have to do any work and we can straight away return -1, since an array of

size 0 won't contain any elements. Also, when we reach the end of an array, that signifies that the array that we now have to search is technically empty, so we can return -1 even in that case. Therefore, our base case will look something like this:

```
if(start == arr.length()):  
    return -1
```

The pseudo-code for this approach is shown on the next page.

```
function solve(arr, x, start):  
    if(start = arr.length()):  
        return -1  
    if(arr[start] = x):  
        return start  
    return solve(arr, x, start + 1)  
  
function firstIndex(arr, x):  
    return solve(arr, x, 0)
```