Last-index-recursion

May 18, 2020

0.1 Problem statement

Given an array `arr` and a target element `target`, find the last index of occurence of `target` in `arr` using recursion. If `target` is not present in `arr`, return `-1`.

For example:

1. For `arr = [1, 2, 5, 5, 1, 2, 5, 4]` and `target = 5`, `output = 6`
2. For `arr = [1, 2, 5, 5, 1, 2, 5, 4]` and `target = 7`, `output = -1`

```
In [2]: def last_index(arr, target):
            """
            :param: arr - input array
            :param: target - integer element
            return: int - last index of target in arr
            TODO: complete this method to find the last index of target in arr
            """
            pass
```

Show Solution

```
In [1]: def test_function(test_case):
            arr = test_case[0]
            target = test_case[1]
            solution = test_case[2]
            output = last_index(arr, target)
            if output == solution:
                print("Pass")
            else:
                print("FAIL: Expected", solution, ", but you've got:", output)
```

```
In [6]: arr = [1, 2, 5, 5, 4]
        target = 5
        solution = 3

        test_case = [arr, target, solution]
        test_function(test_case)
```

```
Pass
```

In [7]:
```
arr = [1, 2, 5, 5, 4]
target = 7
solution = -1

test_case = [arr, target, solution]
test_function(test_case)
```

```
Pass
```

In [9]:
```
arr = [91, 19, 3, 8, 9]
target = 91
solution = 0

test_case = [arr, target, solution]
test_function(test_case)
```

```
Pass
```

In [10]:
```
arr = [1, 1, 1, 1, 1, 1]
target = 1
solution = 5

test_case = [arr, target, solution]
test_function(test_case)
```

```
Pass
```