

Duplicate-Number

May 14, 2020

0.0.1 Problem Statement

You have been given an array of length = n . The array contains integers from 0 to $n - 2$. Each number in the array is present exactly once except for one number which is present twice. Find and return this duplicate number present in the array

Example: * arr = [0, 2, 3, 1, 4, 5, 3] * output = 3 (because 3 is present twice)

The expected time complexity for this problem is $O(n)$ and the expected space-complexity is $O(1)$.

```
In [2]: def duplicate_number(arr):
        """
        :param - array containing numbers in the range [0, len(arr) - 2]
        return - the number that is duplicate in the arr
        """
        current_sum = 0
        expected_sum = 0
        for item in arr:
            current_sum += item

        for num in range(len(arr)-1):
            expected_sum += num
        return current_sum - expected_sum == 0
        pass
```

Hide Solution

```
In [ ]: # Solution
        """
        Notice carefully that
        1. All the elements of the array are always non-negative
        2. If array length = n, then elements would start from 0 to (n-2), i.e. Natural numbers
        3. There is only SINGLE element which is present twice.

        Therefore let's find the sum of all elements (current_sum) of the original array, and
        find the sum of first (n-2) Natural numbers (expected_sum).

        Trick:
        The second occurrence of a particular number (say `x`) is actually occupying the space
```

that would have been utilized by the number (n-1). This leads to:
 $current_sum = 0 + 1 + 2 + 3 + \dots + (n-2) + x$
 $expected_sum = 0 + 1 + 2 + 3 + \dots + (n-2)$
 $current_sum - expected_sum = x$
Tada!!! :)

```
'''
def duplicate_number(arr):
    current_sum = 0
    expected_sum = 0

    # Traverse the original array in the forward direction
    for num in arr:
        current_sum += num

    # Traverse from 0 to (length of array-1) to get the expected_sum
    # Alternatively, you can use the formula for sum of an Arithmetic Progression to get

    # The argument of range() functions are:
    # starting index [OPTIONAL], ending index (non exclusive), and the increment/decrement
    # It means that if the array length = n, loop will run from 0 to (n-2)
    for i in range(len(arr) - 1):
        expected_sum += i

    # The difference between the
    return current_sum - expected_sum
```

```
In [16]: def test_function(test_case):
        arr = test_case[0]
        solution = test_case[1]
        output = duplicate_number(arr)
        if output == solution:
            print("Pass")
        else:
            print("Fail")
```

```
In [17]: arr = [0, 0]
        solution = 0

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

Pass

```
In [18]: arr = [0, 2, 3, 1, 4, 5, 3]
        solution = 3
```

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

Pass

```
In [19]: arr = [0, 1, 5, 4, 3, 2, 0]
        solution = 0
```

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

Pass

```
In [20]: arr = [0, 1, 5, 5, 3, 2, 4]
        solution = 5
```

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

Pass