Return-Subsets

May 18, 2020

0.0.1 Problem Statement

Given an integer array, find and return all the subsets of the array. The order of subsets in the output array is not important. However the order of elements in a particular subset should remain the same as in the input array.

Note: - An empty set will be represented by an empty list. - If there are repeat integers, each occurrence must be treated as a separate entity.

Example 1

```
arr = [9, 9]

output = [[],
          [9],
          [9],
          [9, 9]]
```

Example 2

```
arr = [9, 12, 15]

output =  [[],
           [15],
           [12],
           [12, 15],
           [9],
           [9, 15],
           [9, 12],
           [9, 12, 15]]
```

```
In [3]: def subsets(arr):
            """
            :param: arr - input integer array
            Return - list of lists (two dimensional array) where each list represents a subset
            TODO: complete this method to return subsets of an array
            """
            pass
```

Show Solution

In [26]:
```
def test_function(test_case):
    arr = test_case[0]
    solution = test_case[1]

    output = subsets(arr)

    output.sort()
    solution.sort()

    if output == solution:
        print("Pass")
    else:
        print("Fail")
```

In [27]:
```
arr = [9]
solution = [[], [9]]

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

```
Pass
```

In [28]:
```
arr = [5, 7]
solution = [[], [7], [5], [5, 7]]
test_case = [arr, solution]
test_function(test_case)
```

```
Pass
```

In [29]:
```
arr = [9, 12, 15]
solution = [[], [15], [12], [12, 15], [9], [9, 15], [9, 12], [9, 12, 15]]

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

```
Pass
```

In [30]:
```
arr = [9, 8, 9, 8]
solution = [[],
[8],
[9],
[9, 8],
[8],
[8, 8],
[8, 9],
[8, 9, 8],
```

```
        [9],
        [9, 8],
        [9, 9],
        [9, 9, 8],
        [9, 8],
        [9, 8, 8],
        [9, 8, 9],
        [9, 8, 9, 8]]

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

```
Pass
```