Generating subarrays using recursion

Difficulty Level : Medium • Last Updated : 18 Nov, 2021

Given an array, generate all the possible subarrays of the given array using recursion.

Examples:

```
Input : [1, 2, 3]
Output : [1], [1, 2], [2], [1, 2, 3], [2, 3], [3]

Input : [1, 2]
Output : [1], [1, 2], [2]
```

Recommended: Please try your approach on ***{IDE}*** first, before moving on to the solution.

We have discussed iterative program to generate all subarrays. In this post, recursive is discussed.

Approach: We use two pointers **start** and **end** to maintain the starting and ending point of the array and follow the steps given below:

Below is the implementation of the above approach.

C++

```
// C++ code to print all possible subarrays
// for given array using recursion

#include <iostream>
# include <vector>
using namespace std;

// Recursive function to print all possible subarrays
// for given array
void printSubArrays(vector<int> arr, int start, int end)
{
    // Stop if we have reached the end of the array
    if (end == arr.size())
        return;

    // Increment the end point and start from 0
    else if (start > end)
        printSubArrays(arr, 0, end + 1);

    // Print the subarray and increment the starting point
    else
    {
        cout << "[";
        for (int i = start; i < end; i++){
```

```
    return;
}

int main()
{
   vector<int> arr = {1, 2, 3};
   printSubArrays(arr, 0, 0);
   return 0;
}
```

Java

```
// Java code to print all possible subarrays
// for given array using recursion

class solution
{

// Recursive function to print all possible subarrays
// for given array
static void printSubArrays(int []arr, int start, int end)
{
    // Stop if we have reached the end of the array
    if (end == arr.length)
        return;
```

```
    {
        System.out.print("[");
        for (int i = start; i < end; i++){
            System.out.print(arr[i]+", ");
        }

        System.out.println(arr[end]+"]");
        printSubArrays(arr, start + 1, end);
    }

    return;
}

public static void main(String args[])
{
int []arr = {1, 2, 3};
printSubArrays(arr, 0, 0);

}
}
```

Python3

```
# Python3 code to print all possible subarrays
# for given array using recursion

# Recursive function to print all possible subarrays
```

```
    # Increment the end point and start from 0
    elif start > end:
        return printSubArrays(arr, 0, end + 1)

    # Print the subarray and increment the starting
    # point
    else:
        print(arr[start:end + 1])
        return printSubArrays(arr, start + 1, end)

# Driver code
arr = [1, 2, 3]
printSubArrays(arr, 0, 0)
```

C#

```
// C# code to print all possible subarrays
// for given array using recursion
using System;

class GFG
{

    // Recursive function to print all
    // possible subarrays for given array
    static void printSubArrays(int []arr,
```

```
            return;

        // Increment the end point
        // and start from 0
        else if (start > end)
            printSubArrays(arr, 0, end + 1);

        // Print the subarray and
        // increment the starting point
        else
        {
            Console.Write("[");
            for (int i = start; i < end; i++)
            {
                Console.Write(arr[i]+", ");
            }

            Console.WriteLine(arr[end]+"]");
            printSubArrays(arr, start + 1, end);
        }
        return;
    }

    // Driver code
    public static void Main(String []args)
    {
        int []arr = {1, 2, 3};
        printSubArrays(arr, 0, 0);
    }
}
```

```php
<?php
// PHP code to print all possible
// subarrays for given array using recursion

// Recursive function to print all
// possible subarrays for given array
function printSubArrays($arr, $start, $end)
{
    // Stop if we have reached
    // the end of the array
    if ($end == count($arr))
        return;

    // Increment the end point
    // and start from 0
    else if ($start > $end)
        return printSubArrays($arr, 0,
                              $end + 1);

    // Print the subarray and increment
    // the starting point
    else
    {
    echo "[";
    for($i = $start; $i < $end + 1; $i++)
    {
        echo $arr[$i];
        if($i != $end)
```

```
}

// Driver code
$arr = array(1, 2, 3);
printSubArrays($arr, 0, 0);

// This code is contributed by mits
?>
```

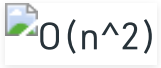
Javascript

```
<script>

// Javascript code to print all possible
// subarrays for given array using recursion

// Recursive function to print all
// possible subarrays for given array
function printSubArrays(arr, start, end)
{

    // Stop if we have reached the end
    // of the array
    if (end == arr.length)
        return;

    // Increment the end point and start
```

```
    else
    {
        document.write("[");
        for(var i = start; i < end; i++)
        {
            document.write( arr[i] + ", ");
        }

        document.write(arr[end] + "]<br>");
        printSubArrays(arr, start + 1, end);
    }
    return;
}

// Driver code
var arr = [ 1, 2, 3 ];
printSubArrays(arr, 0, 0);

// This code is contributed by rutvik_56

</script>
```

Output:

```
[1]
[1, 2]
[2]
```

Time Complexity: 

Previous

Subarray/Substring vs Subsequence and Programs to Generate them

Next

Sum of all Subarrays | Set 1

Article Contributed By :

rituraj_jain
@rituraj_jain

Vote for difficulty

Current difficulty : Medium

Easy | Normal | Medium | Hard | Expert

Improved By : Mithun Kumar, rituraj_jain, SURENDRA_GANGWAR, Rajput-Ji, rutvik_56, sankarsingh, jaskeerat10000

Article Tags : Picked, subarray, Technical Scripter 2018, Arrays, Python, Recursion, Technical Scripter

Practice Tags : Arrays, Recursion

GeeksforGeeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org