

Ace your JavaScript Interview. [Get my ebook](#). 100 solved Javascript, 20 solved React, & 2 frontend system design questions (**1160+ copies sold**). Get a [Free preview](#).



Advertisements

Print all subarrays with a given sum k in an array

Posted on [December 22, 2018](#) | by [Prashant Yadav](#)

Posted in [Algorithms](#), [Arrays](#) | Tagged [Easy](#)

An algorithm to print all subarrays with a given sum k in an [array](#).

Example

Input:

```
[3,4,-7,1,3,3,1,-4]
k = 7
```

Output:

```
[3, 4]
[3, 4, -7, 1, 3, 3]
[1, 3, 3]
[3, 3, 1]
```

Copy

Note: Here we are going to find the consecutive subarrays.

Practically
prepare for
your
JavaScript
interview

[JavaScript
Revision](#)

[JavaScript-
Concept Based
Problems](#)

[Data Structures](#)

[Algorithms](#)

[Machine
Coding](#)

[Web
Fundamentals](#)

Advertisements

Brute Force Method (Naive Approach) $O(n^2)$ to print all the subarrays with given sum

Implementation

- We are going to traverse and sum all the subarrays of the given array and check if they equal to the given sum k.
- If they are equal then we are going to print them.
- Everything will be written in [ES6](#).

[Copy](#)

```
function printSubArrays(arr, k){
    //get the size the of the array
    let length = arr.length;

    //traverse through the array
    for(let i = 0; i < length; i++){
        //temp variables to store the sum and elements
        let tempArr = [];
        let sum = 0;

        //traverse through the every next element after i
        for(let j = i; j < length; j++){
            sum += arr[j];
            tempArr.push(arr[j]);

            //if sum is equal to k then print the array.
            if(sum === k){
                console.log(tempArr);
            }
        }
    }
}

printSubArrays([3,4,-7,1,3,3,1,-4], 7);
```

[Copy](#)**Output:**

```
(2) [3, 4]
(6) [3, 4, -7, 1, 3, 3]
(3) [1, 3, 3]
(3) [3, 3, 1]
//How it works
/*
    length = arr.length = 8;
    Loop
    i = 0; i < 8; i++
    tempArr = [];
    sum = 0;
    Loop
    j = i = 0; j < 8; j++
    sum += arr[j] = 0 + 3 = 3;
    tempArr.push(arr[j]) = [3];
    if(3 === 7){
        //condition fails so continue
    }

    j = i = 1; j < 8; j++
    sum += arr[j] = 3 + 4 = 7;
    tempArr.push(arr[j]) = [3, 4];
    if(7 === 7){
        console.log([3, 4]);
    }

    //continue this till the last element in the inner loop and then check again
    through next element in parents loop
*/
```

Time complexity: $O(N^2)$.

Space complexity: $O(N)$.

Time and Space complexity

- We traversing twice with inner loop, so Time complexity is $O(n^2)$.
- We are storing the array and in worst case all the items of array can equal to the sum, so Space complexity is $O(n)$.

Prepare for your **JavaScript Interview** practically on each Interview rounds and grab that job.

BEGIN LEARNING

Recommended Posts:

[Valid Palindrome II](#)

[FizzBuzz program in javascript](#)

[Given an unsorted array of integers find a pair with given sum in it](#)

[Check if given binary tree is full.](#)

[Print the left view of a binary tree](#)

[Find the largest sum of contiguous subarray.](#)

[How to copy array in javascript](#)

[Swap two numbers without temp variables](#)

[Diagonal traversal of binary tree](#)

[Program to check palindrome linked list](#)

[Prev](#)

[Next](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Start typing...

Name*

Name

Email*

Email

POST COMMENT

[About Us](#) [Contact Us](#) [Privacy Policy](#) [Advertise](#)



Handcrafted with ♥ somewhere in **Mumbai**

© 2023 [LearnersBucket](#) | [Prashant Yadav](#)

