

 Article marked as read.

Subarray with Zero Sum

Given an array of positive and negative numbers, find if there is a subarray (of size at-least one) with 0 sum.

Examples :

Input: {4, 2, -3, 1, 6}

Output: true

Explanation:

There is a subarray with zero sum from index 1 to 3.

Input: {4, 2, 0, 1, 6}

Output: true

Explanation :

The third element is zero. A single element is also a sub-array.

Input: {-3, 2, 3, 1, 6}

Output: false

A **simple solution** is to consider all subarrays one by one and check the sum of every subarray. We can run two loops: the outer loop picks a starting point i and the inner loop tries all subarrays starting from i (See this for implementation). The time complexity of this method is $O(n^2)$.



Dash



All



Articles



Videos



Problems



Quiz



Contest

<< Prev

Next >>

We can also **use hashing**. The idea is to iterate through the array and for every element `arr[i]`, calculate the sum of elements from 0 to `i` (this can simply be done as `sum += arr[i]`). If the current sum is seen before or not, we can check if it is a zero-sum array. Hashing is used to store the sum values so that we can quickly check if the current sum is seen before or not.

✓ Article marked as read.

Example :

```
arr[] = {1, 4, -2, -2, 5, -4, 3}
```

If we consider all prefix sums, we can notice that there is a subarray with 0 sum when :

- 1) Either a prefix sum repeats or
- 2) Or prefix sum becomes 0.

Prefix sums for above array are:

1, 5, 3, 1, 6, 2, 5

Since prefix sum 1 repeats, we have a subarray with 0 sum.

Following is implementation of the above approach.

C++

Java

```
// A Java program to find
// if there is a zero sum subarray
```



Dash



All



Articles



Videos



Problems



Quiz



Contest



 Article marked as read.

Dash



All



Articles



Videos



Problems



Quiz



Contest



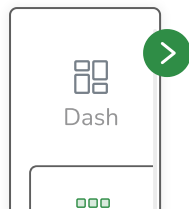
```
import java.util.HashSet;
import java.util.Set;

class ZeroSumSubarray
{
    // Returns true if arr[]
    // has a subarray with zero sum
    static Boolean subArrayExists(int arr[])
    {
        // Creates an empty hashset hs
        Set<Integer> hs = new HashSet<Integer>();

        // Initialize sum of elements
        int sum = 0;

        // Traverse through the given array
        for (int i = 0; i < arr.length; i++)
        {
            // Add current element to sum
            sum += arr[i];

            // Return true in following cases
            // a) Current element is 0
            // b) sum of elements from 0 to i is 0
            // c) sum is already present in hash set
            if (arr[i] == 0
                || sum == 0
                || hs.contains(sum))
                return true;
        }
    }
}
```

 Article marked as read.

Dash

000

Courses

Tutorials

Jobs

Practice

Upcoming
Contests
Articles
Videos
Problems
Quiz
Contest

«

»

```
// Add sum to hash set  
hs.add(sum);  
}
```

```
// We reach here only when there is  
// no subarray with 0 sum
```

```
// Driver code  
public static void main(String arg[])  
{  
    int arr[] = { -3, 2, 3, 1, 6 };  
    if (subArrayExists(arr))  
        System.out.println(  
            "Found a subarray with 0 sum");  
    else  
        System.out.println("No Such Sub Array Exists!");  
}
```

Output

No Such Sub Array Exists!

Time Complexity of this solution can be considered as $O(n)$ under the assumption that we have good hashing function that allows insertion and retrieval operations in $O(1)$ time.

Space Complexity: $O(n)$. Here we required extra space for unordered_set to insert

✓ Article marked as read.

Marked as Read



 Report An Issue

If you are facing any issue on this page. Please let us know.

