Find if there is a subarray with 0 sum

Given an array of positive and negative numbers, find if there is a subarray with 0 sum.

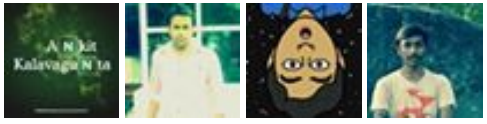
Examples:

```
Input: {4, 2, -3, 1, 6}
Output: true
There is a subarray with zero sum from index 1 to 3.

Input: {4, 2, 0, 1, 6}
Output: true
There is a subarray with zero sum from index 2 to 2.

Input: {-3, 2, 3, 1, 6}
Output: false
There is no subarray with zero sum.
```

We strongly recommend to minimize the browser and try this yourself first.

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). Time complexity of this method is $O(n^2)$.

We can also **use hashing**. The idea is to iterate through the array and for every element arr[i], calculate sum of elements form 0 to i (this can simply be done as sum += arr[i]). If the current sum has been seen before, then there is a zero sum array. Hashing is used to store the sum values, so that we can quickly store sum and find out whether the current sum is seen before or



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion

not.

Following is Java implementation of the above approach.

```
// A Java program to find if there is a zero sum subarray
import java.util.HashMap;

class ZeroSumSubarray {

    // Returns true if arr[] has a subarray with sero sum
    static Boolean printZeroSumSubarray(int arr[])
    {
        // Creates an empty hashMap hM
        HashMap<Integer, Integer> hM = new HashMap<Integer, 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 map
            if (arr[i] == 0 || sum == 0 || hM.get(sum) != null)
               return true;

            // Add sum to hash map
            hM.put(sum, i);
        }

        // We reach here only when there is no subarray with 0 sum
        return false;
    }

    public static void main(String arg[])
    {
        int arr[] = {4, 2, -3, 1, 6};
        if (printZeroSumSubarray(arr))
            System.out.println("Found a subarray with 0 sum");
        else
            System.out.println("No Subarray with 0 sum");
    }
```

Geometric Algorithms

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing
Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

```
}
```

Output:

```
Found a subarray with 0 sum
```

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.

Exercise:
Extend the above program to print starting and ending indexes of all subarrays with 0 sum.

This article is contributed by **Chirag Gupta**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1

- Find the number of zeroes
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3
- Sort n numbers in range from 0 to n^2 – 1 in linear time

Writing code in comment? Please use **ideone.com** and share the link here.

23 Comments GeeksforGeeks

Sort by Newest

AlienOnEarth • 8 days ago

Another approach with o(n) time complexity and o(1) space complexity.

void findZeroSumSubarray(int arr[], int n)

{

int j=0,i=0;

int cur_sum = 0;

int found = 0;

while(i<n) {="" cur_sum="cur_sum" +="" arr[i];="" slide="" window="" if="" curr
while(j<i="" &&="" cur_sum="">0)

{

cur_sum = cur_sum - arr[j];

Recent Comments

kzs please provide solution for the problem...
Backtracking | Set 2 (Rat in a Maze) · 3 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...
Root to leaf path sum equal to a given number · 28 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...
Count trailing zeroes in factorial of a number · 30 minutes ago

newCoder3006 If the array contains negative numbers also. We...
Find subarray with given sum · 54 minutes ago

newCoder3006 Code without using while loop. We can do it...
Find subarray with given sum · 1 hour ago

Sanjay Agarwal You can also use the this method:...
Count trailing zeroes in factorial of a number · 1 hour ago

j++;

see more

1 ^ | v • Reply • Share ›

YoYoHoneySingh • a month ago

I think the logic is simply explained like this: Keep storing sum till index "i". for a
that of "i", this means all the elements between "i" index and "j" index add up to
"j".

1 ^ | v • Reply • Share ›

Sidharth • a month ago

Hi, How does the third condition proves that there might be a zero sum subarr

^ | v • Reply • Share ›

Guest • a month ago

Could anyone please describe how do we come up with the below logic ?

If the current sum has been seen before, then there is a zero sum array.

Shall we apply the same logic to find the given sum(x) subarray in an array ?

1 ^ | v • Reply • Share ›

Srinivas • a month ago

I don't think this solution will work. and the brute force approach time complexi

^ | v • Reply • Share ›

AlienOnEarth ➚ Srinivas • a month ago

can you please provide reason

^ | v • Reply • Share ›

I guess I didn't go thru the solution properly. It works fine. My ba

Reply • Share ›

Venu Gopal AlienOnEarth • a month ago

It is clear that there is subset {-4, -2, 6} for
arr[] = {-4, -2, -3, 1, 6}, but this code gives wrong o/p..
http://ideone.com/FcUWve check this. otherwise if the above c
then it should be mentioned in the problem statement... correct

Reply • Share ›

Kartik Venu Gopal • a month ago

Please take a closer look at the problem statement. It is

Reply • Share ›

alien • a month ago

Hi Geeksforgeeks,

I have another O(n) time and space solution for this problem.

int sumZero(int arr[], int n)

{

int i =0;

int start =0;

int sum = 0;

while(i<n) {="" sum="sum" +="" arr[i];="" while(sum="">0 && start<i) {="" sum
}="" if(sum="=" 0)="" {="" printf("start:="" %d,="" i:="" %d",start,i);="" return;="
i++;="" }="" printf("no="" match="" found");="" }="" i="" request="" you="" to="
it="" in="" the="" solution.="">

GeeksforGeeks Mod ➚ alien • a month ago

alie, could you please post your code on ideone.com and share the link

vrg • a month ago

how is negative index handled?

for ex:
If Input: {-3, 2, 3, 1, 6}

During 1st iteration when i=0
sum=-3 and how/where is -3 stored in hashtable?

hM.put(sum, i);

hm.put(-3,0) where is this stored?

Ankur gupta • a month ago

Hash map method doesn't work for example1 also.

Kartik ➚ Ankur gupta • a month ago

It seems to be working fine. See http://ideone.com/pKPxnU

Rajat Sadh • a month ago

http://ideone.com/FswPty

Ravi ➚ Rajat Sadh • a month ago

Doesn't work for many cases, ex {3, -2, -1, -3, 6}. There is a subarray,

Reply • Share ›

zzer Ravi • a month ago

it checks if sum==0, so the code will find it

1 Reply • Share ›

GOPI GOPINATH • a month ago

is the hashmap package existing ?? or we need to write our own hashmap an

Reply • Share ›

Ravi GOPI GOPINATH • a month ago

it is part of standard java

Reply • Share ›

GOPI GOPINATH Ravi • a month ago

forgot about the case of the letter 'h' in hashmap...got it !

Reply • Share ›

Babu • a month ago

Any specific reason why we are starting from i=1 . int i = 1 not i=0

Reply • Share ›

GeeksforGeeks Mod Babu • a month ago

Babu, that seemed to be typo. Thanks for pointing this out. We have c

Reply • Share ›

Praveen GeeksforGeeks • 3 days ago

Could you put both the Alien's methods on ideone and place the

Reply • Share ›