https://course.acciojob.com/idle?question=2ee2a709-fb2f-4acd-b328-a7a74a556edb

- EASY
- Max Score: 20 Points
-
-
-

Largest Subarray with 0 Sum

Given an array arr[] of length N, find the length of the longest sub-array with a sum equal to 0.

A subarray is a contiguous part of the array.

Complete the given function.

Input Format:

First line contains integer N

Second line contains n integers arr[i].

Output Format:

Print the longest subarray length with a sum equal to 0.

EXAMPLE 1:

Input:

```
8
15 -2  2 -8 1 7 10 23
```

Output:

```
5
```

Explanation:

The longest sub-array with elements summing up-to 0 is {-2, 2, -8, 1, 7}

EXAMPLE 2:

Input:

```
3
1 2 3
```

Output:

```
0
```

Explanation:

There is no subarray with 0 sum

constraints:

```
-100000 <= nums.length <= 100000
```

```
0 <= nums[i] <= 100000
```

Topic Tags

- Arrays

My code

```
// in java
import java.util.HashMap;
import java.io.*;
import java.util.*;

class Solution {
   public int maxLen(int arr[])
        {
                // Write your code here
                int len=0;
                int sum=0;
                HashMap<Integer,Integer>hm=new HashMap<>();
                hm.put(0,-1);
                for(int i=0;i<arr.length;i++)
                        {
                                sum+=arr[i];
                                if(hm.containsKey(sum))
                                {
                                int l=i-hm.get(sum);
                                        if(len<l)
                                                len=l;
                                }
                                else hm.put(sum,i);
                        }
                return len;
        }
}

class Main {

        public static void main(String arg[])
        {
                Scanner sc = new Scanner(System.in);
                int n = sc.nextInt();
        int[] nums = new int[n];
        for (int i = 0; i < n; i++)
        {
            nums[i] = sc.nextInt();
        }
    Solution Obj = new Solution();
        System.out.println(Obj.maxLen(nums));
    }
}
```