**Find the length of largest subarray with 0 sum**

<https://www.geeksforgeeks.org/find-the-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.

**Examples:**

***Input:****arr[] = {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}*

***Input:****arr[] = {1, 2, 3}****Output:****0****Explanation:****There is no subarray with 0 sum*

***Input:****arr[] = {1, 0, 3}****Output:****1****Explanation:****The longest sub-array with elements summing up-to 0 is {0}*

**Time Complexity:** O(N2)  
**Auxiliary Space:** O(1)

def maxLen(arr):

    # initialize result

    max\_len = 0

    # pick a starting point

    for i in range(len(arr)):

        # initialize sum for every starting point

        curr\_sum = 0

        # try all subarrays starting with 'i'

        for j in range(i, len(arr)):

            curr\_sum += arr[j]

            # if curr\_sum becomes 0, then update max\_len

            if curr\_sum == 0:

                max\_len = max(max\_len, j-i + 1)

    return max\_len

# Driver's code

if \_\_name\_\_ == "\_\_main\_\_":

# test array

    arr = [15, -2, 2, -8, 1, 7, 10, 13]

    # Function call

    print ("Length of the longest 0 sum subarray is % d" % maxLen(arr))

**Time Complexity:** O(N)  
**Auxiliary Space:** O(N)

def maxLen(arr):

    # NOTE: Dictionary in python is

    # implemented as Hash Maps

    # Create an empty hash map (dictionary)

    hash\_map = {}

    # Initialize result

    max\_len = 0

    # Initialize sum of elements

    curr\_sum = 0

    # Traverse through the given array

    for i in range(len(arr)):

        # Add the current element to the sum

        curr\_sum += arr[i]

        if curr\_sum == 0:

            max\_len = i + 1

        # NOTE: 'in' operation in dictionary

        # to search key takes O(1). Look if

        # current sum is seen before

        if curr\_sum in hash\_map:

            max\_len = max(max\_len, i - hash\_map[curr\_sum])

        else:

            # else put this sum in dictionary

            hash\_map[curr\_sum] = i

    return max\_len

# Driver's code

if \_\_name\_\_ == "\_\_main\_\_":

    # test array

    arr = [15, -2, 2, -8, 1, 7, 10, 13]

    # Function call

    print("Length of the longest 0 sum subarray is % d" % maxLen(arr))