560. Subarray Sum Equals K

10 February 2022 09:58 AM

Given an array of integers `nums` and an integer `k`, return *the total number of continuous subarrays whose sum equals to* `k`.

Example 1:

```
Input: nums = [1,1,1], k = 2
Output: 2
```

Example 2:

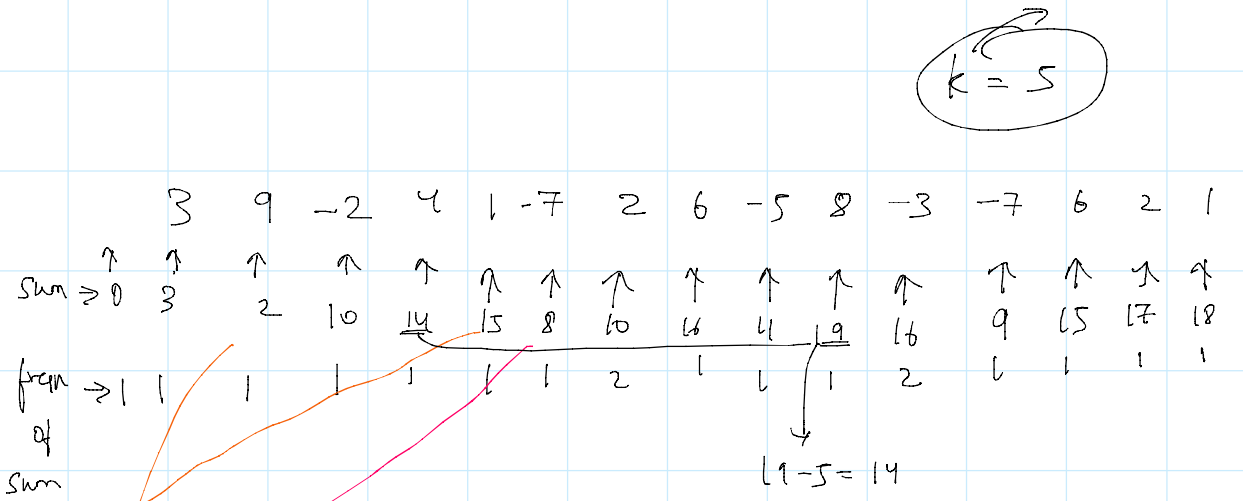
```
Input: nums = [1,2,3], k = 3
Output: 2
```

Brute force

```
count = 0
for (start = 0, ⟶ start++)
    for(end = start+1 ⟶ end++)
        int sum = 0
        for(int i = start, i < end, i++)
            sum += nums[i]
        if (sum == k)
            count++
return count
```

T.C ⇒ $O(n^3)$ S.C ⇒ $O(1)$

Using hash map –



Check whether a is already present or not for e.g. now k=5 and at arr ⇒ 9 the sum is 12. So now 12 − 5 = 7, and 7 is not present in the hashmap (sum) so it's not the subarray.

Here 15 − 5 = 10, 10 is already present in the hash map so one subarray of k=5 is (4,1)

Here 8 − 5 = 3, 3 is already there so one more subarray

→ (4,1)

→ (9, −2, 4, 1, −7)

→ 1 −7 2 6 −5 8

→ 8 −3

→ 4 1 −7 2 6 −5 8 −3 −7 6

→ 6 −5 8 −3 −7 6

→ −2 4 1 −7 2 6 −5 8 −3 −7 6 2

before inserting in the hashmap, we will check if the number is present or not if yes add in subarray

(num − k)