Subarray sum divisible by k

974. Subarray Sums Divisible by K

Medium 6012 252 Add to List Share

Given an integer array `nums` and an integer `k`, return *the number of non-empty **subarrays** that have a sum divisible by* `k`.

A **subarray** is a **contiguous** part of an array.

Example 1:

```
Input: nums = [4,5,0,-2,-3,1], k = 5
Output: 7
Explanation: There are 7 subarrays with a sum divisible by k = 5:
[4, 5, 0, -2, -3, 1], [5], [5, 0], [5, 0, -2, -3], [0], [0, -2, -3], [-2, -3]
```

Example 2:

```
Input: nums = [5], k = 9
Output: 0
```

Brute force

Since we have to find Sum of all subArray divisble by K

→ Nested loop

[4, 5, 0, -2, -3, 1] K = 5

i ↗ ↗ j

At i = 0

j = 0
sum = 0 + 4 ⇒ 4
if (sum % K == 0) false

j = 1
sum = 4 + 5 = 9
if (sum % K == 0) false

j = 2
sum = 9 + 0 = 9
if (sum % K == 0) false

j = 3
sum = 9 + (-2) = 7
if (sum % K == 0) false

j = 4
sum = 7 - 3 = 4
if (sum % K == 0) false

j = 5
sum = 4 + 1 = 5
if (sum % K == 0) True cnt++

→ i++, a[i] = 5

j = 0
sum = 0 + 5
if (sum % K == 0) True cnt++

so on

T.C - O(n²)

S.C - O(1)

TLE

Optimal approach

Hashing :-

subarray sum divisible by K

arr = [$a_0, a_1, a_2, \ldots a_i, \ldots a_{n-1}, a_n$], K

Need

a[i:j] s.t sum(a[i:j]) % K = 0

Calculate prefix sum for array

F = [$a_0, a_1, a_2 \ldots a_i, \ldots a_j, \ldots a_n$]

S_i

$r_i = S_i \% K$

S_j

$r_j = S_j \% K$

if $r_j = r_i$

$S_j \% K - S_i \% K = 0$

$(S_j - S_i) \% K = 0$

so let us take example :-

0	1	2	3	4	5
4	5	0	-2	-3	1

	i=0	i=1	i=2	i=3	i=4	i=5
Prefix	4	4+5 = 9	9+0 = 9	9-2 = 7	7-3 = 4	4+1 = 5
Find -	4%5 = 4	9%5 = 4	9%5 = 4	7%4 = 3	4%5 = 4	5%5 = 0
	False	True	True	False	True	
insert :	4	4++	4++	3	0++	

Rem	Freq
0	1
4	1 2 3
3	

cnt = mp[Find]

= 1 + 2 = 3 + 3 = 6 + 1

= 7 ans.

```cpp
class Solution {
public:
    int subarraysDivByK(vector<int>& nums, int k) {
        unordered_map<int,int> mpp;
        mpp[0]=1;
        int sum=0,cnt=0;
        for(int i=0;i<nums.size();i++){
            sum+=nums[i];
            int findRem =sum%k ;
            if(findRem <0) findRem+=k;
            if(mpp.find(findRem)!=mpp.end()) cnt+=mpp[findRem];
            mpp[findRem]++;
        }
        return cnt;
    }
};
```