523. Continuous Subarray Sum

Add to List

Description SubmissionsSolutions

- Total Accepted: 5863
- Total Submissions: 27280
- Difficulty: Medium
- Contributors:xuehaohu

Given a list of **non-negative** numbers and a target **integer** k, write a function to check if the array has a continuous subarray of size at least 2 that sums up to the multiple of **k**, that is, sums up to n*k where n is also an **integer**.

Example 1:

```
Input: [23, 2, 4, 6, 7],  k=6

Output: True

Explanation: Because [2, 4] is a continuous subarray of size 2 and sums up t
o 6.
```

Example 2:

```
Input: [23, 2, 6, 4, 7],  k=6

Output: True

Explanation: Because [23, 2, 6, 4, 7] is an continuous subarray of size 5 an
d sums up to 42.
```

Note:

1. The length of the array won't exceed 10,000.
2. You may assume the sum of all the numbers is in the range of a signed 32-bit integer.

Subscribe to see which companies asked this question

```
class Solution {
public:
    bool checkSubarraySum(vector<int>& nums, int k) {
        int n = nums.size();
        vector<int> sum(n+1,0);
        for(int i=1;i<=n;i++)
        {
            sum[i] = sum[i-1]+nums[i-1];
            for(int j=0;j<i-1;j++)
            {
                int diff = sum[i]-sum[j];
                if(diff==0 && k==0) return true;
                else if (k==0) continue;
                else if(diff%k==0) return true;
            }
        }
        return false;
    }
```

};