

Problem 523: Continuous Subarray Sum

Problem Information

Difficulty: Medium

Acceptance Rate: 31.08%

Paid Only: No

Tags: Array, Hash Table, Math, Prefix Sum

Problem Description

Given an integer array `nums` and an integer `k`, return `true` if `nums` has a **good subarray** or `false` otherwise.

A **good subarray** is a subarray where:

* its length is **at least two**, and * the sum of the elements of the subarray is a multiple of `k`.

Note that:

* A **subarray** is a contiguous part of the array. * An integer `x` is a multiple of `k` if there exists an integer `n` such that `x = n * k`. `0` is **always** a multiple of `k`.

Example 1:

Input: `nums = [23, 2, 4, 6, 7]`, `k = 6` **Output:** `true` **Explanation:** `[2, 4]` is a continuous subarray of size 2 whose elements sum up to 6.

Example 2:

Input: `nums = [23, 2, 6, 4, 7]`, `k = 6` **Output:** `true` **Explanation:** `[23, 2, 6, 4, 7]` is a continuous subarray of size 5 whose elements sum up to 42. 42 is a multiple of 6 because $42 = 7 * 6$ and 7 is an integer.

Example 3:

****Input:**** nums = [23,2,6,4,7], k = 13 ****Output:**** false

****Constraints:****

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 109` * `0 <= sum(nums[i]) <= 231 - 1` * `1 <= k <= 231 - 1`

Code Snippets

C++:

```
class Solution {  
public:  
    bool checkSubarraySum(vector<int>& nums, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
public boolean checkSubarraySum(int[] nums, int k) {  
  
}  
}
```

Python3:

```
class Solution:  
    def checkSubarraySum(self, nums: List[int], k: int) -> bool:
```