6329. Make K-Subarray Sums Equal

My Submissions (/contest/biweekly-contest-101/problems/make-k-subarray-sums-equal/submissions/) | Back to Contest (/contest/biweekly-contest-101/)

You are given a **0-indexed** integer array `arr` and an integer `k`. The array `arr` is circular. In other words, the first element of the array is the next element of the last element, and the last element of the array is the previous element of the first element.

You can do the following operation any number of times:

- Pick any element from `arr` and increase or decrease it by `1`.

Return *the minimum number of operations such that the sum of each **subarray** of length* `k` *is equal*.

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

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

```
Input: arr = [1,4,1,3], k = 2
Output: 1
Explanation: we can do one operation on index 1 to make its value equal to 3.
The array after the operation is [1,3,1,3]
- Subarray starts at index 0 is [1, 3], and its sum is 4
- Subarray starts at index 1 is [3, 1], and its sum is 4
- Subarray starts at index 2 is [1, 3], and its sum is 4
- Subarray starts at index 3 is [3, 1], and its sum is 4
```

Example 2:

```
Input: arr = [2,5,5,7], k = 3
Output: 5
Explanation: we can do three operations on index 0 to make its value equal to 5 and two operations on index 3 to make its value
The array after the operations is [5,5,5,5]
- Subarray starts at index 0 is [5, 5, 5], and its sum is 15
- Subarray starts at index 1 is [5, 5, 5], and its sum is 15
- Subarray starts at index 2 is [5, 5, 5], and its sum is 15
- Subarray starts at index 3 is [5, 5, 5], and its sum is 15
```

Constraints:

- $1 <= k <= arr.length <= 10^5$
- $1 <= arr[i] <= 10^9$

JavaScript ▾   ⟨⟩ ↻ ⚙

```javascript
const makeSubKSumEqual = (a, k) => medianGreedy(a, k)

const gcd = (a, b) => b == 0 ? a : gcd(b, a % b);
const medianGreedy = (a, k) => {
    k = gcd(a.length, k);
    let res = 0;
    for (let i = 0; i < k; i++) {
        let b = [];
        for (let j = i; j < a.length; j += k) b.push(a[j]);
        b.sort((x, y) => x - y);
        let mid = b[b.length >> 1];
        for (const x of b) res += Math.abs(x - mid);
    }
    return res;
};
```

☐ **Custom Testcase**   (Use Example Testcases)

⏵ Run   ☁ Submit

Submission Result: Accepted (/submissions/detail/926065132/) ❓   (More Details ❯ (/submissions/detail/926065132/))

Share your acceptance!

Copyright © 2023 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

🇺🇸 United States (/region)