5987. Minimum Difference in Sums After Removal of Elements

My Submissions (/contest/biweekly-contest-71/problems/minimum-difference-in-sums-after-removal-of-elements/submissions/)

Back to Contest (/contest/biweekly-contest-71/)

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

You are given a **0-indexed** integer array `nums` consisting of `3 * n` elements.

You are allowed to remove any **subsequence** of elements of size **exactly** `n` from `nums`. The remaining `2 * n` elements will be divided into two **equal** parts:

- The first `n` elements belonging to the first part and their sum is $\text{sum}_{\text{first}}$.
- The next `n` elements belonging to the second part and their sum is $\text{sum}_{\text{second}}$.

The **difference in sums** of the two parts is denoted as $\text{sum}_{\text{first}} - \text{sum}_{\text{second}}$.

- For example, if $\text{sum}_{\text{first}} = 3$ and $\text{sum}_{\text{second}} = 2$, their difference is `1`.
- Similarly, if $\text{sum}_{\text{first}} = 2$ and $\text{sum}_{\text{second}} = 3$, their difference is `-1`.

Return *the* ***minimum difference*** *possible between the sums of the two parts after the removal of* `n` *elements*.

Example 1:

```
Input: nums = [3,1,2]
Output: -1
Explanation: Here, nums has 3 elements, so n = 1.
Thus we have to remove 1 element from nums and divide the array into two equal parts.
- If we remove nums[0] = 3, the array will be [1,2]. The difference in sums of the two parts will be 1 - 2 = -1.
- If we remove nums[1] = 1, the array will be [3,2]. The difference in sums of the two parts will be 3 - 2 = 1.
- If we remove nums[2] = 2, the array will be [3,1]. The difference in sums of the two parts will be 3 - 1 = 2.
The minimum difference between sums of the two parts is min(-1,1,2) = -1.
```

Example 2:

```
Input: nums = [7,9,5,8,1,3]
Output: 1
Explanation: Here n = 2. So we must remove 2 elements and divide the remaining array into two parts containing two elements
If we remove nums[2] = 5 and nums[3] = 8, the resultant array will be [7,9,1,3]. The difference in sums will be (7+9) - (1+3
To obtain the minimum difference, we should remove nums[1] = 9 and nums[4] = 1. The resultant array becomes [7,5,8,3]. The c
It can be shown that it is not possible to obtain a difference smaller than 1.
```

Constraints:

- `nums.length == 3 * n`
- $1 <= n <= 10^5$
- $1 <= nums[i] <= 10^5$

JavaScript

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var minimumDifference = function(nums) {

};
```