1848. Minimum Distance to the Target Element

My Submissions (/contest/weekly-contest-239/problems/minimum-distance-to-the-target-element/submissions/)

Back to Contest (/contest/weekly-contest-239/)

Given an integer array `nums` **(0-indexed)** and two integers `target` and `start`, find an index `i` such that `nums[i] == target` and `abs(i - start)` is **minimized**. Note that `abs(x)` is the absolute value of `x`.

Return `abs(i - start)`.

It is **guaranteed** that `target` exists in `nums`.

User Accepted:	5281
User Tried:	5483
Total Accepted:	5382
Total Submissions:	8578
Difficulty:	Easy

Example 1:

```
Input: nums = [1,2,3,4,5], target = 5, start = 3
Output: 1
Explanation: nums[4] = 5 is the only value equal to target, so the answer is abs(4 - 3) = 1.
```

Example 2:

```
Input: nums = [1], target = 1, start = 0
Output: 0
Explanation: nums[0] = 1 is the only value equal to target, so the answer is abs(0 - 0) = 1.
```

Example 3:

```
Input: nums = [1,1,1,1,1,1,1,1,1,1], target = 1, start = 0
Output: 0
Explanation: Every value of nums is 1, but nums[0] minimizes abs(i - start), which is abs(0 - 0) = 0.
```

Constraints:

- `1 <= nums.length <= 1000`
- `1 <= nums[i] <= 10^4`
- `0 <= start < nums.length`
- `target` is in `nums`.

Discuss (https://leetcode.com/problems/minimum-distance-to-the-target-element/discuss)

Java

```
class Solution {
    public int getMinDistance(int[] nums, int target, int start) {

    }
}
```