Problem 1848: Minimum Distance to the Target Element

Problem Information

Difficulty: Easy
Acceptance Rate: 54.23%
Paid Only: No
Tags: Array

Problem Description

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`.

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) = 0.

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] <= 104` * `0 <= start < nums.length` * `target` is in `nums`.

Code Snippets

C++:

```
class Solution {
public:
int getMinDistance(vector<int>& nums, int target, int start) {

}
};
```

Java:

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

}
}
```

Python3:

```
class Solution:
def getMinDistance(self, nums: List[int], target: int, start: int) -> int:
```