# 题目

给你一个整数数组nums（下标从0开始计数）以及两个整数target和start，请你找出一个下标i，满足nums[i] == target且abs(i - start) 最小化。注意：abs(x)表示x的绝对值。

返回 abs(i - start) 。

题目数据保证 target 存在于 nums 中。

示例 1：

输入：nums = [1,2,3,4,5], target = 5, start = 3

输出：1

解释：nums[4] = 5 是唯一一个等于 target 的值，所以答案是 abs(4 - 3) = 1 。

示例 2：

输入：nums = [1], target = 1, start = 0

输出：0

解释：nums[0] = 1 是唯一一个等于 target 的值，所以答案是 abs(0 - 0) = 0 。

示例 3：

输入：nums = [1,1,1,1,1,1,1,1,1,1], target = 1, start = 0

输出：0

解释：nums 中的每个值都是 1 ，但 nums[0] 使 abs(i - start) 的结果得以最小化，所以答案是 abs(0 - 0) = 0 。

提示：

1 <= nums.length <= 1000

1 <= nums[i] <= 10^4

0 <= start < nums.length

target 存在于 nums 中

# 分析

## 方法一：模拟

思路：

代码：

class Solution {

public:

int getMinDistance(vector<int>& nums, int target, int start) {

int res = nums.size();

for (int i = 0; i < nums.size(); ++i){

if (nums[i] == target){

res = min(res, abs(i - start));

}

}

return res;

}

};

复杂度分析

时间复杂度：O(n)，即为遍历数组的时间复杂度。

空间复杂度：O(1)。