

Description | Editorial | Solutions | Submissions

1509. Minimum Difference Between Largest and Smallest Value in Three Moves

Medium

Topics

Companies

Hint

You are given an integer array `nums`.

In one move, you can choose one element of `nums` and change it to **any value**.

Return *the minimum difference between the largest and smallest value of `nums` after performing at most three moves*.

Example 1:

Input: `nums = [5,3,2,4]`

Output: `0`

Explanation: We can make at most 3 moves.

In the first move, change 2 to 3. `nums` becomes `[5,3,3,4]`.

In the second move, change 4 to 3. `nums` becomes `[5,3,3,3]`.

In the third move, change 5 to 3. `nums` becomes `[3,3,3,3]`.

After performing 3 moves, the difference between the minimum and maximum is $3 - 3 = 0$.







Example 2:

Input: `nums = [1,5,0,10,14]`

Output: `1`

Explanation: We can make at most 3 moves.

In the first move, change 5 to 0. `nums` becomes `[1,0,0,10,14]`.

 2.5K   177 |   

7 Online

Code


C++ • Auto

```
1 class Solution {
2 public:
3     int minDifference(vector<int>& nums) {
4
5     }
6 };
```

SavedLn 1, Col 1

You need to [Login / Sign up](#) to run or submit

☒ Testcase

 Test Result