

给定一个非空整数数组，找到使所有数组元素相等所需的最小移动数，其中每次移动可将选定的一个元素加1或减1。您可以假设数组的长度最多为10000。

**例如：**

**输入：**

[1,2,3]

**输出：**

2

**说明：**

只有两个动作是必要的（记得每一步仅可使其中一个元素加1或减1）：

[1,2,3] => [2,2,3] => [2,2,2]

思路，将数组排序，取中间的数字，这样，整个数组被划分为2块，左边小于等于中间数，右边大于等于中间数，这时若以中间数作为移动值，和最小。应为若是值比中间数大，则左边和中间需全部增加1，右边最多减少左边加上中间个移动1，其值最好也是步数不变，同理小于。

**代码：**

```
class Solution {
public:
    int minMoves2(vector<int>& nums) {
        sort(nums.begin(),nums.end());
        int i=nums[nums.size()/2];
        int num=0;
        for(int k:nums){
            num+=abs(k-i);
        }
        return num;
    }
};
```