1. 题目
2. 题干

给定一个长度为 n 的**非空**整数数组，找到让数组所有元素相等的最小移动次数。每次移动将会使 n - 1 个元素增加 1。

1. 示例

输入:

[1,2,3]

输出:

3

解释:

只需要3次移动（注意每次移动会增加两个元素的值）：

[1,2,3] => [2,3,3] => [3,4,3] => [4,4,4]

二． 题解

（一）思路

一开始按照排序的方法来做，选取前n-1个数加1，结果可想而知超时！！！后来翻阅官方题解的数学解法，简直是醍醐灌顶！！！

题目要求每次移动就要将除了某一个数以外的n-1个数增加1，这种做法等价与将该数减1，其他n-1个数不做任何改动。就是这么简单的思想，我们不需要重复地对数组进行排序，只需要找出数组中的最小值，让其他的数减小到最小值即可。计算其他值减小到最小值的次数和就是最终的答案！！！

（二） 代码实现

Java：

class Solution {

    public int minMoves(int[] nums) {

        //数学法

        //让除了一个数以外的所有数都加1，等价于只让该数减1， 艹了

        //因此，我们只需要知道最小的数，求出其他数减小到该数的移动次数即可

        int n = nums.length;

        int min = nums[0];

        int sum = 0;

        for(int i = 0; i < n; ++i){

            min = Math.min(min, nums[i]);

        }

        for(int i = 0; i < n; ++i){

            sum += nums[i] - min;

        }

        return sum;

    }

}