453 Minimum Moves to Equal Array Elements

2018年4月11日 18:49

Given a **non-empty** integer array of size *n*, find the minimum number of moves required to make all array elements equal, where a move is incrementing *n* - 1 elements by 1.

Example:

Input:

[1,2,3]

Output:

3

Explanation:

Only three moves are needed (remember each move increments two elements):

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

来自 <https://leetcode.com/problems/minimum-moves-to-equal-array-elements/description/>

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。每次移动可以使 *n* - 1 个元素增加 1。

解析：

Incrementing all but one is equivalent to decrementing that one. So let's do that instead. How many single-element decrements to make all equal? No point to decrementing below the current minimum。

来自 <https://leetcode.com/problems/minimum-moves-to-equal-array-elements/discuss/93822/Simple-one-liners>

Solution for Python3:

```python
class Solution:
    def minMoves(self, nums):
        """
        :type nums: List[int]
        :rtype: int
        """
        return sum(nums) - min(nums) * len(nums)
```

Solution for C++:

```cpp
class Solution {
public:
    int minMoves(vector<int>& nums) {
        return accumulate(begin(nums), end(nums), 0L) - nums.size() * *min_element(begin(nums), end(nums));
    }
};
```