

## 移动石子直到连续

三枚石子放置在数轴上，位置分别为  $a$ ,  $b$ ,  $c$ 。

每一回合，你可以从两端之一拿起一枚石子（位置最大或最小），并将其放入两端之间的任一空闲位置。形式上，假设这三枚石子当前分别位于位置  $x$ ,  $y$ ,  $z$  且  $x < y < z$ 。那么就可以从位置  $x$  或者是位置  $z$  拿起一枚石子，并将该石子移动到某一整数位置  $k$  处，其中  $x < k < z$  且  $k \neq y$ 。

当你无法进行任何移动时，即，这些石子的位置连续时，游戏结束。

要使游戏结束，你可以执行的最小和最大移动次数分别是多少？ 以长度为 2 的数组形式返回

答案: `answer = [minimum_moves, maximum_moves]`

示例 1:

输入:  $a = 1, b = 2, c = 5$

输出: `[1, 2]`

解释: 将石子从 5 移动到 4 再移动到 3，或者我们可以直接将石子移动到 3。

示例 2:

输入:  $a = 4, b = 3, c = 2$

输出: `[0, 0]`

解释: 我们无法进行任何移动。

```
class Solution:
    def numMovesStones(self, a: int, b: int, c: int) -> List[int]:
        #根据题意
        ...

        题目并未规定 a,b,c 之间的大小关系，所以需要先排序，排序后  $a < b < c$ :
        1、若  $c - a == 2$ ，则无法移动
        2、若 a、c 任一与 b 差值小于等于 2，则可一步到位，否则至少两步
        3、最多移动步数为  $c - a - 2$ 
        ...

        #先排序
        tmp=[a,b,c]
        tmp.sort()
        a,b,c=tmp[0],tmp[1],tmp[2]
        #如果存在两两相邻
        if(c-a==2):
            return [0,0]
```

```
#如果一侧相邻，另外一侧不相邻
if c-b<=2 or b-a<=2:
    return [1,c-a-2] #当一侧相邻最小移动次数一定为1 此时情况为：只需要移动另外一侧直接移动到中间位置的相邻位置
#两侧都不相邻，两侧都需要移动
else:
    return [2,c-a-2]#当两侧均不相邻最小移动次数一定为2 此时情况为：两侧需要移动到中间位置的相邻位置 共两次
```