1. 题目
2. 题干

数轴上放置了一些筹码，每个筹码的位置存在数组 chips 当中。

你可以对 任何筹码 执行下面两种操作之一（不限操作次数，0 次也可以）：

将第 i 个筹码向左或者右移动 2 个单位，代价为 0。

将第 i 个筹码向左或者右移动 1 个单位，代价为 1。

最开始的时候，同一位置上也可能放着两个或者更多的筹码。

返回将所有筹码移动到同一位置（任意位置）上所需要的最小代价。

1. 示例

示例 1：

输入：chips = [1,2,3]

输出：1

解释：第二个筹码移动到位置三的代价是 1，第一个筹码移动到位置三的代价是 0，总代价为 1。

示例 2：

输入：chips = [2,2,2,3,3]

输出：2

解释：第四和第五个筹码移动到位置二的代价都是 1，所以最小总代价为 2。

提示：

1 <= chips.length <= 100

1 <= chips[i] <= 10^9

1. 题解
2. 思路

从题目来看，要想用最小的开销来将所有的筹码移动到一个位置，就要尽量使用开销为0的移动方式，在真的无法继续使用开销0的方式时，再使用开销为1的方式。开销为0的方式要求我们间隔两个格子来移动，那么我们可以将所有的奇数位置的筹码都通过这种方式移动到一个奇数位置Odd， 将所有的偶数位置的筹码都移动到跟Odd位置相邻的偶数Even位置，这样移动的开销就为0。此时Odd位置中的就有position数组中所有奇数个数的筹码，Even位置就有position数组中所有偶数个数的筹码。之后，就必须使用开销为1的移动方式了，两个位置中，哪个位置的筹码数少就移动哪个位置的筹码到另一个位置中，问题解决！！！！

1. 代码实现

Java：

class Solution {

    public int minCostToMoveChips(int[] position) {

        int even = 0, odd = 0;

        for(int pos : position){

            if(pos % 2 == 0){

                even++;

            }else{

                odd++;

            }

        }

        return Math.min(even, odd);

    }

}