

玩筹码

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。

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。

解题思路

问题转换：

偶数次移动是无需代价的，那么久存在对于奇数或者偶数位置对应元素都移动偶数次，带来结果就是可以让他们相邻就差 1

剩下要决定的就是是奇数位置之和 移动 到偶数位置之和，这里就是取最小的

举个例子

[0, 1, 2, 3, 4, 5, 6]

移动偶数次后,比如偶数移到 0,奇数移到 1,代价为 0, 那么就是

[12, 9, 0, 0, 0, 0, 0]

这里需要移动一个单位,12<9 那么移动 9 到 12, 最后是 [21, 0,

0, 0, 0, 0, 0], 代价是 9

```
class Solution {
public:
    int minCostToMoveChips(vector<int>& position) {
        int sum1=0,sum2=0;
        for(int i=0;i<position.size();i++)
        {
            if(position[i]%2==0)
            {
                sum1++;//偶数统计
            }
            else
            {
                sum2++;//奇数统计
            }
        }
        return min(sum1,sum2);//取二者最小的实现代价为 1 的操作
    }
};
```