

题目描述：

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从 $0 \sim N$ 的箱子，每个箱子上面贴有箱子中藏有金币的数量。

从金币数量中选出一个数字集合，并销毁贴有这些数字的每个箱子，如果能销毁一半及以上的箱子，则返回这个数字集合的最小大小。

输入描述：

一个数字字符串，数字之间使用逗号分隔，例如：6,6,6,6,3,3,3,1,1,5

字符串中数字的个数为偶数，并且个数 ≥ 1 ， ≤ 100000 ；每个数字 ≥ 1 ， ≤ 100000 ；

输出描述：

这个数字集合的最小大小，例如：2

示例 1

输入：

1,1,1,1,3,3,3,6,6,8

输出：

2

说明：

选择集合 $\{1,8\}$ ，销毁后的结果数组为 $[3,3,3,6,6]$ ，长度为 5，长度为原数组的一半。

大小为 2 的可行集合还有 $\{1,3\}, \{1,6\}, \{3,6\}$ 。

选择 $\{6,8\}$ 集合是不可行的，它销毁后的结果数组为 $[1,1,1,1,3,3,3]$ ，新数组长度大于原数组的二分之一。

示例 2

输入：

2,2,2,2

输出：

1

说明：

我们只能选择集合{2}，销毁后的结果数组为空。

```
import sys
import collections

def del_half_num(nums):
    if len(set(nums)) == len(nums):
        return len(nums) // 2
    if len(set(nums)) == 1:
        return 1

    counter = collections.Counter(nums).most_common()
    current_arr_len = len(nums)
    del_count = 0

    for tuple_each in counter:
        if current_arr_len <= (len(nums) // 2):
            return del_count
        else:
            current_arr_len -= tuple_each[1]
            del_count += 1
    return del_count

if __name__ == "__main__":
    input = sys.stdin.readline().strip().split(',')
    print(del_half_num(input))
```