

Java - 阿里巴巴找黄金宝箱(II) - 一贫如洗的樵夫阿里巴巴在去砍柴的路上

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 ， ≤ 1000000 ；每个数字 ≥ 1 ， ≤ 1000000 ;

输出描述:

这个数字集合的最小大小，例如: 2

补充说明:

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 java.util.*;

// 注意类名必须为 Main，不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String c = sc.nextLine();
        String[] split = c.split(",");
        int[] ints = new int[split.length];
        Set set = new HashSet();
        for(int i=0;i<=split.length-1;i++)
        {
            ints[i] = Integer.parseInt(split[i]);
            set.add(ints[i]);
        }
        int size = set.size();
        int[] intsl = new int[size];
        Object[] objects = set.toArray();
        for(int j=0;j<objects.length;j++)
        {
            for(int k=0;k<ints.length;k++)
```

```

        {
            if(Integer.parseInt(objects[j].toString())==ints[k])
            {
                intsl[j]++;
            }
        }
    }
    List<Integer> list = new ArrayList();
    for (int z=0;z<intsl.length;z++)
    {
        list.add(intsl[z]);
    }
    Collections.sort(list);
    int n = 0;
    for(int t=list.size()-1;t>=0;t--)
    {
        n +=list.get(t);
        if(n>=split.length/2)
        {
            System.out.println(list.size()-t);
            break;
        }
    }
}
}

```