

Java方向编程题答案

day20

[编程题]26026-微信红包

链接: <https://www.nowcoder.com/questionTerminal/fbcf95ed620f42a88be24eb2cd57ec54>

【题目解析】

无

【解题思路】

数据可能有不存在超过红包一半的, 这个时候就要输出0; 即该金额的数目必须 大于 红包数一半, 不能小于等于。

如果一个数出现次数超过一半了, 排序过后, 必然排在中间, 则最后遍历整个排序后的数组查看是否符合条件。

【示例代码】

```
import java.util.*;
public class Gift {
    public int getValue(int[] gifts, int n) {
        Arrays.sort(gifts); // 数组排序, 采用Arrays的sort方法
        int ans = gifts[n/2]; // 理论上超过半数的数字
        int num = 0;
        // 遍历数组, 进行统计
        for(int i = 0; i < gifts.length; i++) {
            if(gifts[i] == ans) {
                num++;
            }
        }
        return num <= n/2 ? 0 : ans;
    }
}
```

//最优解

```
import java.util.*;
public class Gift {
    public int getValue(int[] gifts, int n) {
        if(gifts.length < n) return 0;
        if(gifts.length == 0) return 0;
        int count = 0, temp = 0;
        for(int i = 0; i < n; i++)
        {
```

```

        if(count==0)
        {
            temp=gifts[i];
            count=1;
        }
        else{
            if(temp==gifts[i])
                count++;
            else
                count--;
        }
    }
    int size=0;
    for(int i=0;i<n;i++){
        if(temp==gifts[i])
            size++;
    }
    return (size>n/2)?temp:0;
}
}

```

[编程题]24913-链表分割

链接: <https://www.nowcoder.com/questionTerminal/0e27e0b064de4eacac178676ef9c9d70>

【题目解析】

无

【解题思路】

该题目的目的是将pHead链表中的值以小于x的在链表前面，大于等于x的在链表后面。

- 定义两个链表，第1个链表表示小于x的节点；第2个链表表示大于等于x的节点，遍历pHead链表，进行与x的比较，就可以将链表的节点分开
- 将第1个链表和第2个链表连接在一起即可

【示例代码】

```

import java.util.*;

/*
public class ListNode {
    int val;
    ListNode next = null;

    ListNode(int val) {
        this.val = val;
    }
}*/
public class Partition {

    public ListNode partition(ListNode pHead, int x) {

```

```

    if (pHead == null || pHead.next == null) {
        return pHead;
    }
    ListNode cur = pHead;

    //定义2个链表, 此处Shead Bhead两个头指针
    ListNode Shead = new ListNode(-1);
    ListNode Bhead = new ListNode(-1);

    ListNode Stmp = Shead;
    ListNode Btmp = Bhead;
    while (cur != null) {
        if (cur.val < x) { //值小于x的节点
            Stmp.next = new ListNode(cur.val);
            Stmp = Stmp.next;
        } else { //值大于等于x的节点
            Btmp.next = new ListNode(cur.val);
            Btmp = Btmp.next;
        }
        cur = cur.next;
    }
    //第1个链表的头
    cur = Shead;
    //循环遍历找到第1个链表的尾
    while (cur.next != null && cur.next.val != -1) {
        cur = cur.next;
    }
    //cur的next指向第2个节点的next(非头节点)
    //相当于将第1个链表和第2个链表连接
    cur.next = Bhead.next;
    return Shead.next; //返回第1个节点next(不含头节点)
}
}

```