链滴

linkedlist 应用之 golang 解 leetcode203

作者：xhaoxiong

原文链接：https://ld246.com/article/1538667593978

来源网站：链滴

0、main函数测试

```
package main

import (
    "important_test/data_structer/linkList"
    "fmt"
)

func main() {

    var s = []interface{}{25, 25, 25, 25, 27, 28, 29, 30}

    link := linkList.SliceToList(s)
    fmt.Println(link.ToString())

    link.NormalRemove(25)
    fmt.Println(link.ToString())
}


//leetcode题解递归法
type ListNode struct {
    Val  int
    Next *ListNode
}

func removeElements(head *ListNode, val int) *ListNode {
    if head == nil {
        return nil
    }
    head.Next = removeElements(head.Next, val)
    if head.Val == val {
        return head.Next
    } else {
        return head
    }
}
```

1、 使用普通删除法 带虚拟头节点

```
//正常去除节点
func (this *LinkList) NormalRemove(elem interface{}) {

    prev := this.dummyHead

    for prev.Next != nil {
        cur := prev.Next
        fmt.Println(cur.Elem)
        if cur.Elem != elem {
            prev = prev.Next
        }
        prev.Next = cur.Next
```

```
    }

    return
}
```

2、递归法

```
func (this *LinkList) RemoveRepeat(val interface{}) {
    this.RecursiveRemove(this.dummyHead, val)
}

//递归法Remove链表中重复的值
func (this *LinkList) RecursiveRemove(head *node, elem interface{}) *node {
    if head == nil {
        return nil
    }

    head.Next = this.RecursiveRemove(head.Next, elem)

    if head.Elem == elem {
        return head.Next
    } else {
        return head
    }
}
```

具体代码