

题目 从尾到头打印链表

考点 链表 热点指数 118871 通过率 24.44%

具体题目

输入一个链表，按链表值从尾到头的顺序返回一个ArrayList。

有三种思路，第一就是利用栈先入后出的特性完成，第二就是存下来然后进行数组翻转。第三是利用递归。

```
import java.util.ArrayList;
import java.util.Collections;
public class Solution {
    public ArrayList<Integer> printListFromTailToHead(ListNode listNode) {
        ArrayList<Integer> list = new ArrayList<Integer>();

        while(listNode != null){
            list.add(listNode.val);
            listNode = listNode.next;
        }

        Collections.reverse(list); //使用Collections的reverse方法，直接将list反转
        return list;
    }
}
```

最佳代码：代码思路借助栈，遍历的时候入栈，由于数据结构中栈的特点是先进后出，所以遍历的过程中压栈，推栈，完了弹栈加到ArrayList中。有两个容易出错的地方：第一，第一次测试用例，{}返回[], null是null，而[]是new ArrayList()但是没有数据。第二，遍历stack用的方法是！stk.isEmpty()方法，而不是for循环size遍历。。 /**

- public class ListNode {
- int val;
- ListNode next = null; *
- ListNode(int val) {
- this.val = val;
- }
- } **/

```
import java.util.Stack;
import java.util.ArrayList;
public class Solution {
    public ArrayList<Integer>
    printListFromTailToHead(ListNode listNode) {
        if(listNode == null){
            ArrayList list = new ArrayList();
            return list;
        }
        Stack<Integer> stk = new Stack<Integer>();
        while(listNode != null){
            stk.push(listNode.val);
            listNode = listNode.next;
        }
        ArrayList<Integer> arr = new ArrayList<Integer>();
        while(!stk.isEmpty()){
            arr.add(stk.pop());
        }
        return arr;
    }
}
```

