**难度: 简单**

输入一个链表的头节点，从尾到头反过来返回每个节点的值（用数组返回）。

**示例 1：**

**输入：**head = [1,3,2]

**输出：**[2,3,1]

**思路1:**

1). 先遍历节点,把所有值,放到ArrayList中

2). 创建一个和ArrayList长度一样的数组

3). ArrayList倒着遍历存储到数组中

|  |
| --- |
| public int[] reversePrint(ListNode head) {  //创建一个临时节点  ListNode temp = head;  //创建一个ArrayList集合  List<Integer> list = new ArrayList<Integer>();  //遍历  while(temp != null) {  //存储  list.add(temp.val);  //下一个节点  temp = temp.next;  }  //长度  int len = list.size();  //创建数组  int[] res = new int[len];  //下标  int index = 0;  //倒着存储  for (int i = len -1; i >= 0; i--) {  res[index++] = list.get(i);  }  return res;  } |

思路2:

1). 先遍历计算整个链表的长度

2). 创建一个和链表长度相等的数组

3). 重写遍历链表,并且倒着存储到数组中

|  |
| --- |
| public int[] reversePrint2(ListNode head) {  //创建一个临时节点  ListNode temp = head;    //统计  int count = 0;    while(temp != null) {  count++;  temp = temp.next;  }    temp = head;    //创建数组  int[] res = new int[count];    //倒着存储  while(temp != null) {  res[--count] = temp.val;  temp = temp.next;  }  return res;  } |