

剑指06 从头到尾打印链表

利用递归做到反向访问链表而不改变链表结构。利用访问顺序push到vector中

```
1 //链表反转的写法肌肉记忆!!
2 //input: ListNode* head
3
4 //普通迭代反转链表: _____
5 ListNode *pre = NULL;
6 ListNode *cur = head;
7 while (cur != NULL) {
8     // 反转链表四步大法
9     ListNode *next = cur->next;
10    cur->next = pre;
11    pre = cur;
12    cur = next;
13 }
14
15
16 //递归反转链表法(c++ vector输出): _____(非常巧妙)
17 //时间空间最高
18 vector<int> reversePrint(ListNode* head) {
19     if(!head)
20         return {};
21     vector<int> a=reversePrint(head->next);
22     a.push_back(head->val);
23     return a;
24 }
```

```
1 利用中间栈方法。
2 //使用algorithm算法中的reverse反转res
3 reverse(res.begin(),res.end());
4
```