# 算法

## 链表

1. 反转单链表

* 从原链表的头部一个一个取节点并插入到新链表的头部，两个链表，一个新链表，一个原链表

|  |
| --- |
| ListNode\* ReverseList(ListNode\* pHead) {  ListNode\* newh = NULL;  for(ListNode\* p = pHead; p; )//p为工作指针  {  ListNode\* tmp = p -> next;//temp保存下一个结点  p -> next = newh;  newh = p;  p = tmp;  }  return newh;  } |

* 每次都将原第一个结点之后的那个结点放在新的表头后面。一个链表即可

## 字符串

1. 旋转字符串

给定一个字符串，要求把字符串前面的若干个字符移动到字符串的尾部，如把字符串“abcdef”前面的2个字符'a'和'b'移动到字符串的尾部，使得原字符串变成字符串“cdefab”。请写一个函数完成此功能，要求对长度为n的字符串操作的时间复杂度为 O(n)，空间复杂度为 O(1)。

* 三步反转法

abc => cba，def => fed，cbafed => defabc