链表题目总览

# LeetCode上Linked List所有题目

## 总览叙述

2018-10-05目前LeetCode一共算法题目909道，其中做了75道，35道Easy，30道Medium，10道Hard。

筛选出的LinkedList题目，可以看出，链表题目只有2道Hard题目，所以一般关于链表LinkedList的题目都是简单中等题目，属于基础题目。

## 所有题目如下：

2 Add Two Numbers 29.2% Medium

19 Remove Nth Node From End of List 33.6% Medium

21 Merge Two Sorted Lists 43.3% Easy

23 Merge k Sorted Lists 30.4% **Hard**

24 Swap Nodes in Pairs 41.0% Medium

25 Reverse Nodes in k-Group 33.4% Hard

61 Rotate List 25.5% Medium

82 Remove Duplicates from Sorted List II 31.0% Medium

83 Remove Duplicates from Sorted List 40.9% Easy

86 Partition List 34.9% Medium

92 Reverse Linked List II 32.6% Medium

109 Convert Sorted List to Binary Search Tree 37.5% Medium

138 Copy List with Random Pointer 25.6% Medium

141 Linked List Cycle 34.6% Easy

142 Linked List Cycle II 29.9% Medium

143 Reorder List 28.2% Medium

147 Insertion Sort List 35.1% Medium

148 Sort List 31.9% Medium

160 Intersection of Two Linked Lists 30.7% Easy

203 Remove Linked List Elements 34.3% Easy

206 Reverse Linked List 49.6% Easy

234 Palindrome Linked List 34.3% Easy

237 Delete Node in a Linked List 49.5% Easy

328 Odd Even Linked List 46.5% Medium

369 Plus One Linked List 55.0% Medium

379 Design Phone Directory 38.1% Medium

426 Convert Binary Search Tree to Sorted Doubly Linked List 38.7% Medium

430 Flatten a Multilevel Doubly Linked List 34.3% Medium

445 Add Two Numbers II 47.4% Medium

707 Design Linked List 17.9% Easy

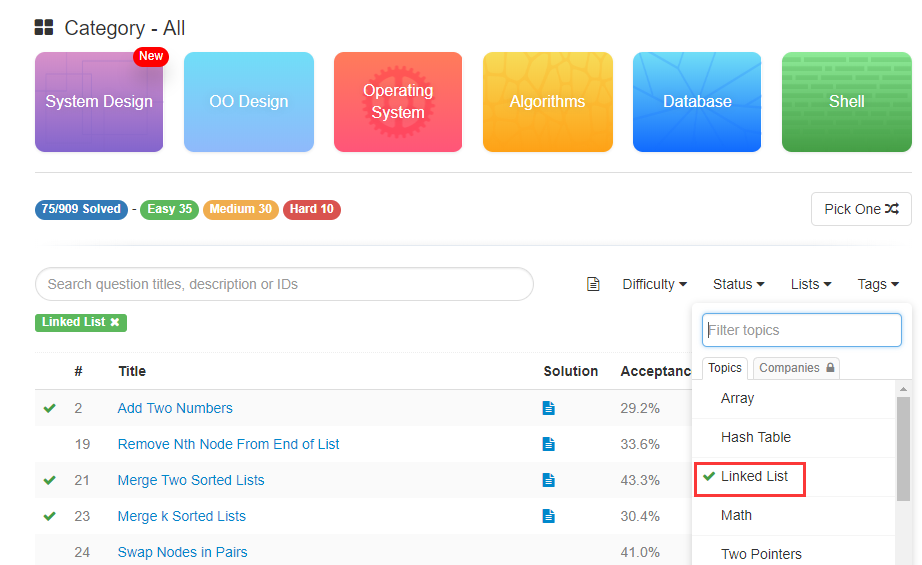
708 Insert into a Cyclic Sorted List 23.7% Medium

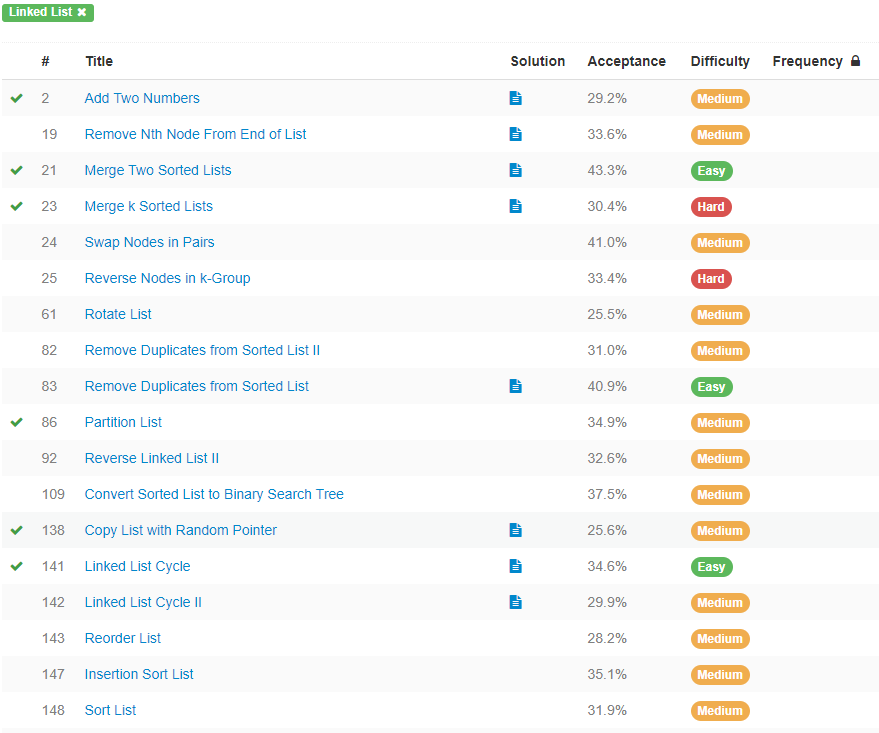
725 Split Linked List in Parts 47.3% Medium

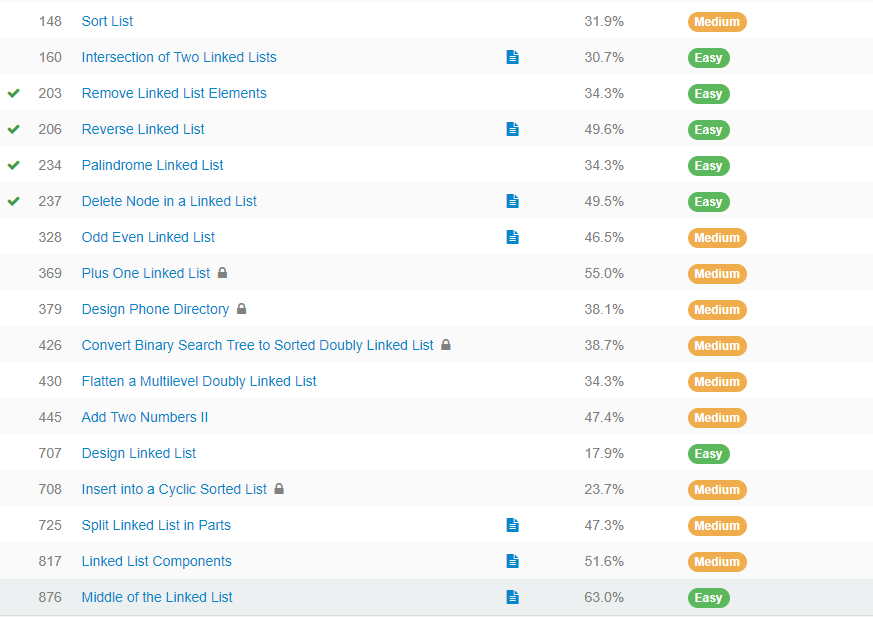
817 Linked List Components 51.6% Medium

876 Middle of the Linked List 63.0% Easy

## 截图如下







# 链表LinkedList的题目类型分类

### 从尾至头打印链表；<栈>

### 查找节点：

查找倒数第K个节点；<双指针ahead和behind>

查找两个链表的第一个公共节点；

### 删除节点：

O(1)复杂度删除指定节点；

删除重复节点：I重复的留一个，II重复的一个也不留。

### 反转链表；

整体翻转，部分翻转。

### 分割链表：

分割成bigger和samller；

### 链表的复制：

复杂链表的复制；

### 链表中的环：

判断链表中是否含有环；<双指针fast,slow，fast走两步，slow走一步>

求环的长度；<相遇节点转一圈即可>

寻找环的入口节点；<方法1：ahead和behind，相差环的长度步><方法2：需要数学证明，两个指针，一指向相遇点，一指向起始点，必定在环入口点相遇>

### 合并链表：

Easy：合并两个链表；(递归法、while循环)

Hard：合并K个链表; (逐个合并法、排序法、分治递归法)