

链表总结篇

链表的理论基础

如下几点：

- 链表的种类主要为：单链表，双链表，循环链表
- 链表的存储方式：链表的节点在内存中是分散存储的，通过指针连在一起。
- 链表是如何进行增删改查的。
- 数组和链表在不同场景下的性能分析。

可以说把链表基础的知识都概括了，但又不像教科书那样的繁琐。

链表经典题目

虚拟头结点

链表操作中一个非常重要的技巧：虚拟头节点。

链表的一大问题就是操作当前节点必须要找前一个节点才能操作。这就造成了，头结点的尴尬，因为头结点没有前一个节点了。

每次对应头结点的情况都要单独处理，所以使用虚拟头结点的技巧，就可以解决这个问题。

用虚拟头结点和没用虚拟头结点的代码，对比一下就会发现，使用虚拟头结点的好处。

链表的基本操作

我们通过设计链表把链表常见的五个操作练习了一遍。

这是练习链表基础操作的非常好的一道题目，考察了：

- 获取链表第index个节点的数值
- 在链表的最前面插入一个节点
- 在链表的最后面插入一个节点
- 在链表第index个节点前面插入一个节点
- 删除链表的第index个节点的数值

可以说把这道题目做了，链表基本操作就OK了，再也不用担心链表增删改查整不明白了。

这里我依然使用了虚拟头结点的技巧，大家复习的时候，可以去看一下代码。

反转链表

讲解了如何反转链表。

因为反转链表的代码相对简单，有的同学可能直接背下来了，但一写还是容易出问题。

反转链表是面试中高频题目，很考察面试者对链表操作的熟练程度。

给出了两种反转的方式，迭代法和递归法。

建议大家先学透迭代法，然后再看递归法，因为递归法比较绕，如果迭代还写不明白，递归基本也写不明白了。

可以先通过迭代法，彻底弄清楚链表反转的过程！

删除倒数第N个节点

我们结合虚拟头结点 和 双指针法来移除链表倒数第N个节点。

链表相交

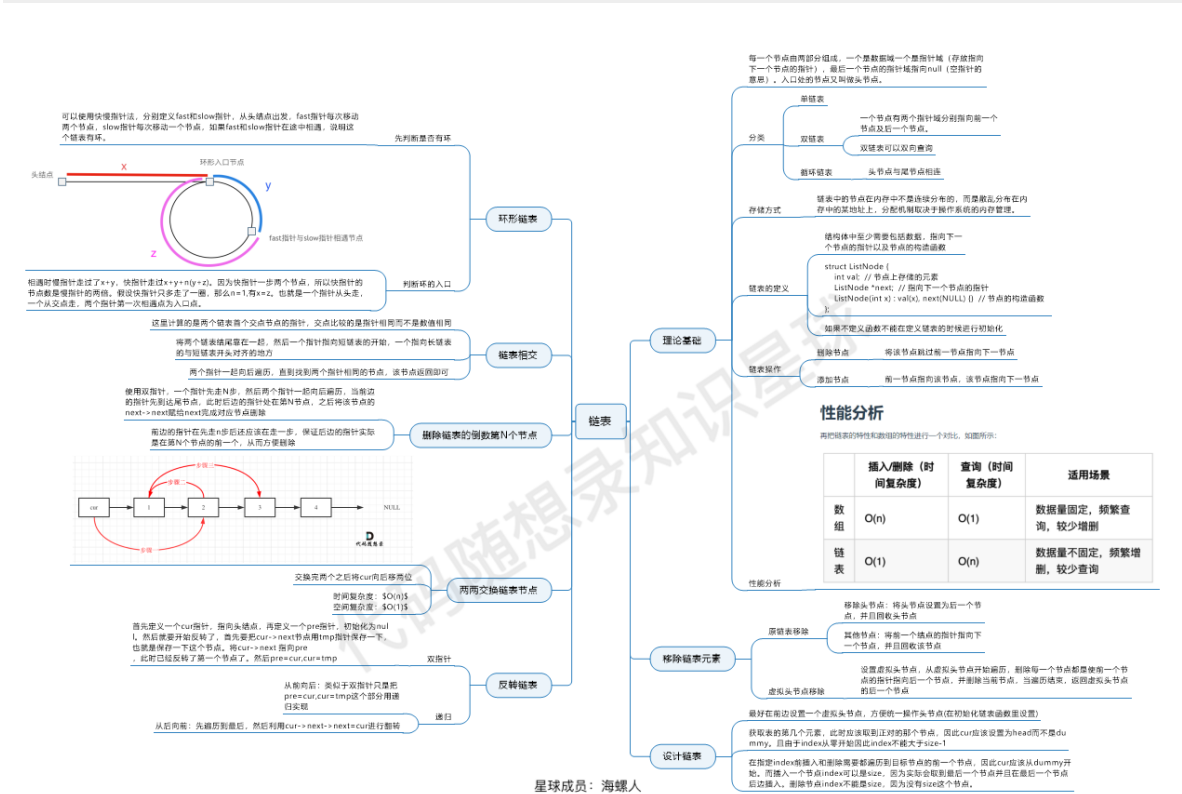
使用双指针来找到两个链表的交点（引用完全相同，即：内存地址完全相同的交点）

环形链表

在链表如何找环，以及如何找环的入口位置。

这道题目可以说是链表的比较难的题目了。但代码却十分简洁，主要在于一些数学证明。

总结



星球成员：海螺人

1. 有否环形，以及环的入口