



BITTIGER

CS102 Top100高频算法设计课

第二节 链表

左程云



版权声明

所有太阁官方网站以及在第三方平台课程中所产生的课程内容，如文本，图形，徽标，按钮图标，图像，音频剪辑，视频剪辑，直播流，数字下载，数据编辑和软件均属于太阁所有并受版权法保护。

对于任何尝试散播或转售BitTiger的所属资料的行为，太阁将采取适当的法律行动。

我们非常感谢您尊重我们的版权内容。

有关详情，请参阅

<https://www.bittiger.io/termsfuse>

<https://www.bittiger.io/termservice>



Copyright Policy

All content included on the Site or third-party platforms as part of the class, such as text, graphics, logos, button icons, images, audio clips, video clips, live streams, digital downloads, data compilations, and software, is the property of BitTiger or its content suppliers and protected by copyright laws.

Any attempt to redistribute or resell BitTiger content will result in the appropriate legal action being taken.

We thank you in advance for respecting our copyrighted content. For more info see <https://www.bittiger.io/termsfuse> and <https://www.bittiger.io/termservice>



简单的介绍单链表和双链表

```
LinkedListNode{  
    T value;  
    LinkedListNode next;  
}  
  
DoubleLinkedListNode{  
    T value;  
    DoubleLinkedListNode last;  
    DoubleLinkedListNode next;  
}
```



常见水题

【水题一】

打印两个链表的公共部分

【水题二】

删除单链表中倒数第K个节点

删除双链表中倒数第K个节点

【水题三】

实现删除链表的中间节点的函数

实现删除链表中位于a/b处节点的函数

【水题四】

分别实现反转单向链表和反转双向链表的函数



常见水题

【水题五】

在单向链表上把第from个节点到第to个节点这一部分进行反转，from和to是整数

【水题六】

实现一个调整单链表的函数，使得每K个节点之间逆序，最后不够的部分不调整

【水题七】

给定一个无序单链表的头节点head，删除值重复的节点（保留一个）

【水题八】

给定一个链表的头节点head和一个整数num，请实现函数将值为num的节点全部删除



常见水题

【水题九】

给定有序环形单链表（最后指回头部） 和一个新节点N， 插入N并保证继续有序

【水题十】

给定两个有序单链表的头节点h1和h2， 请合并两个有序链表并返回大链表头节点

【水题十一】

链表长度为N， 如N为偶数， 前N/2个节点为左区， 后N/2个节点为右区； 如N为奇数， 前N/2个节点为左区， 后N/2+1个节点为右区。 调整链表成L1->R1->L2->R2->...的形式

【水题十二】

给定一个链表中的节点node， 但不给定整个链表的头节点， 如何在链表中删除node？



环形单链表的约瑟夫问题

【题目一】

据说著名犹太历史学家Josephus有过以下故事：在罗马人占领乔塔帕特后，39个犹太人与Josephus及他的朋友躲到一个洞中，39个犹太人决定宁愿死也不要被敌人抓到，于是决定了一个自杀方式，41个人排成一个圆圈，由第1个人开始报数，报数到3的人就自杀，然后再由下一个人重新报1，报数到3的人再自杀，这样依次下去，直到剩下最后一个人时，那个人可以自由选择自己的命运。这就是著名的约瑟夫问题。现在请用单向环形链表描述该结构并呈现整个自杀过程。

输入：一个环形单向链表的头节点head和报数的值m。

返回：最后生存下来的节点，且这个节点自己组成环形单向链表，其他节点都删掉。

【进阶】

如果链表节点数为N，想在时间复杂度为 $O(N)$ 时完成原问题的要求，该怎么实现？



环形单链表的约瑟夫问题 解题思路

【解题点】
待补充！



BITTIGER



判断一个链表是否为回文结构

【题目二】

给定一个链表的头节点head，请判断该链表是否为回文结构。

例如：

1->2->1，返回true。

1->2->2->1，返回true。

123->6->123，返回true。

1->2->3，返回false。



判断一个链表是否为回文结构 解题思路

【方法一】

利用空间大小为 N 的栈，全逆序比对

【方法二】

利用空间大小为 $N/2$ 的栈，折半逆序比对

【方法三】

- 1, 上去调整链表使得左右半区都指向中间节点
- 2, 从两头一步一步走着比对
- 3, 最后把链表调整回来



单链表的partition操作

【题目三】

将单向链表按某值划分成左边小、中间相等、右边大的样子



单链表的partition操作 解题思路

【方法一】

用容器

【方法二】

- 1, 分别设置小于、等于、大于, 三个头部
- 2, 遍历时, 把遇到的节点, 根据条件挂在某个头部之下
- 3, 把三个部分串起来



复制含有随机指针节点的链表

【题目四】

一种特殊的链表节点类描述如下：

```
Node {  
    T value;  
    Node next;  
    Node rand;  
}
```

rand指针是Node类中新增的指针，这个指针可能指向整个链表中的任意一个节点，也可能指向null，给定由这种种类的节点组成的链表头部，请完成该链表的拷贝



复制含有随机指针节点的链表 解题思路

【方法一】
用容器

【方法二】

- 1, 把拷贝的节点放在原始节点的后面, 即, 旧, 新, 旧, 新...
- 2, 利用旧, 新, 旧, 新...的放置方式把新节点之间的rand指针串好
- 3, 把旧的和新的分离出来
- 4, 返回新的链表



两个单链表相交的一系列问题

【题目五】

约定：单链表可能有环，也可能无环

给定两个单链表的头节点head1和head2，这两个链表可能相交，也可能不相交。请实现一个函数，如果两个链表相交，请返回相交的第一个节点；如果不相交，返回null



两个单链表相交的一系列问题 解题思路

【方法一】
用容器

【方法二】
分解成三个问题：

- 1，怎么得到单链表入环的第一个节点，还不能额外空间
快与慢在相交之处，让快跳到头部和慢同步走到再次相交
- 2，怎么得到两个无环单链表的第一个相交节点，还不能额外空间
长的先走，走到和短的长度一样时，同步走
- 3，怎么得到两个有环单链表的第一个相交节点，还不能额外空间
一个一直走，绕回自己之前遇到另一个则相交，反之不相交
- 4，这道题希望好好练，不光是想法很值得学习，对于处理单链表的coding技巧提升也很有帮助



将搜索二叉树转换成双向链表

【题目六】

对二叉树的节点来说，有本身的值域，有指向左孩子和右孩子的两个指针；

对双向链表的节点来说，有本身的值域，有指向上一个节点和下一个节点的指针；

在结构上，两种结构有相似性，现在有一棵搜索二叉树，请将其转换为一个有序的双向链表。



将搜索二叉树转换成双向链表 解题思路

【解题点】

- 1, 时间复杂度可以是 $O(N)$
- 2, 利用返回左右两个节点或链表重连等特殊处理, 可以省掉遍历的代价
- 3, 对练习递归设计很有帮助



设计可以变更的缓存结构

【题目七】

设计一种缓存结构，该结构在构造时确定大小，假设大小为K，并有两个功能：

`set(key,value)`：将记录(key,value)插入该结构

`get(key)`：返回key对应的value值

【要求】

1. `set`和`get`方法的时间复杂度为 $O(1)$ 。
2. 某个key的`set`或`get`操作一旦发生，认为这个key的记录成了最经常使用的
3. 当缓存的大小超过K时，移除最不经常使用的记录（`set`或`get`发生最久远的节点）



【解题点】

1, 做一个双向链表, 实现:

`void addNode(newNode)`

`void moveNodeToTail(node)`

`Node removeHead()`

2, 用两个哈希表建立起原始的key和双向链表节点之间的互相联系

3, 两张哈希表和一个双向链表的配合问题



【题目八】

已知一个流会不断地吐出整数 $1 \sim N$ ，但不一定按顺序吐出，根据以下的描述设计一个结构，支持接受和打印。流会吐出全部的 $1 \sim N$ ，当然也会都打印完，要求整个过程，时间复杂度为 $O(N)$ 。

吐出2，接收2不打印，因为1还没出现；吐出1，接收1并打印1，2；吐出4，接收4而不打印，因为3还没出现；吐出5，接收5而不打印，因为3还没出现；吐出7，接收7而不打印，因为3还没出现；吐出3，接收3并打印3，4，5；吐出9，接收9而不打印，因为6还没出现；吐出8；接收8而不打印，因为6还没出现；吐出6，接收6，并打印6，7，8，



一种消息接收并打印的结构设计 解题思路

【解题点】

- 1, 用哈希表map1, key为一个区间开始消息的编号, value为消息体
- 2, 用哈希表map2, key为一个区间结束消息的编号, value为消息体
- 3, 一个区间内部, 消息体按照消息编号的顺序组成单链表
- 4, 一个编号为i的消息到来时, 先生成只有该消息的一个区间, 即更新map1、map2、建立只有自己的单链表; 然后在map2中查看有没有以i-1结尾的区间, 有的话合并; 然后在map1中查看有没有以i+1开头的区间, 有的话合并; 合并过程包括更新map1、map2以及区间链表的首位重新连接
- 5, 纪录上一次打印的消息编号, 每加入一个新的消息, 看看是否又触发了打印逻辑, 打印时就是单链表的遍历过程

课程项目负责人：**Catherine**

邮件：weiyi@bittiger.io

左程云答疑邮箱：chengyunzuo@gmail.com

微信二维码：



关注微信，获得太阁最新信息

微信：**bit_tiger**

官网：**BitTiger.io**

