

牛客网算法基础入门班

第四课

牛客网最新算法课--基础入门班：针对0基础算法小白，左神手把手讲解入门数据结构和算法原理，结合典型例题深入讲解解题思路，并提供最优解和代码

上课时间：每周六日 09:00——11:00

上课老师：左程云，华科本科，芝加哥大学硕士，曾就职于IBM、百度、GrowingIO、亚马逊，也是牛客网的老师。

牛客网：一个提供海量校招真题及专项练习题，笔经面经，招聘信息，学习资源及交流的平台<https://www.nowcoder.com/>



笔经面经



学习交流

哈希表的简单介绍

- 1) 哈希表在使用层面上可以理解为一种集合结构
- 2) 如果只有key, 没有伴随数据value, 可以使用HashSet结构(C++中叫UnOrderedSet)
- 3) 如果既有key, 又有伴随数据value, 可以使用HashMap结构(C++中叫UnOrderedMap)
- 4) 有无伴随数据, 是HashMap和HashSet唯一的区别, 底层的实际结构是一回事
- 5) 使用哈希表增(put)、删(remove)、改(put)和查(get)的操作, 可以认为时间复杂度为 $O(1)$, 但是常数时间比较大
- 6) 放入哈希表的东西, 如果是基础类型, 内部按值传递, 内存占用就是这个东西的大小
- 7) 放入哈希表的东西, 如果不是基础类型, 内部按引用传递, 内存占用是这个东西内存地址的大小

有关哈希表的原理, 将在提升班“与哈希函数有关的数据结构”一章中讲叙原理

有序表的简单介绍

- 1) 有序表在使用层面上可以理解为一种集合结构
- 2) 如果只有key，没有伴随数据value，可以使用TreeSet结构(C++中叫OrderedSet)
- 3) 如果既有key，又有伴随数据value，可以使用TreeMap结构(C++中叫OrderedMap)
- 4) 有无伴随数据，是TreeSet和TreeMap唯一的区别，底层的实际结构是一回事
- 5) 有序表和哈希表的区别是，有序表把key按照顺序组织起来，而哈希表完全不组织
- 5) 红黑树、AVL树、size-balance-tree和跳表等都属于有序表结构，只是底层具体实现不同
- 6) 放入~~哈希表~~的东西，如果是基础类型，内部按值传递，内存占用就是这个东西的大小
- 7) 放入~~哈希表~~的东西，如果不是基础类型，必须提供比较器，内部按引用传递，内存占用是这个东西内存地址的大小
- 8) 不管是什么底层具体实现，只要是有序表，都有以下固定的基本功能和固定的时间复杂度

有序表的固定操作

- 1) void put(K key, V value): 将一个 (key, value) 记录加入到表中, 或者将key的记录更新成value。
- 2) V get(K key): 根据给定的key, 查询value并返回。
- 3) void remove(K key): 移除key的记录。
- 4) boolean containsKey(K key): 询问是否有关于key的记录。
- 5) K firstKey(): 返回所有键值的排序结果中, 最左 (最小) 的那个。
- 6) K lastKey(): 返回所有键值的排序结果中, 最右 (最大) 的那个。
- 7) K floorKey(K key): 如果表中存入过key, 返回key; 否则返回所有键值的排序结果中, key的前一个。
- 8) K ceilingKey(K key): 如果表中存入过key, 返回key; 否则返回所有键值的排序结果中, key的后一个。

以上所有操作时间复杂度都是 $O(\log N)$, N 为有序表含有的记录数

有关有序表的原理, 将在提升班“有序表详解”一章中讲叙原理

题目四

单链表的节点结构

```
Class Node<V>{  
    V value;  
    Node next;  
}
```

由以上结构的节点依次连接起来所形成的链叫单链表结构。

双链表的节点结构

```
Class Node<V>{  
    V value;  
    Node next;  
    Node last;  
}
```

由以上结构的节点依次连接起来所形成的链叫双链表结构。

单链表和双链表结构只需要给定一个头部节点head，就可以找到剩下的所有的节点。

反转单向和双向链表

【题目】 分别实现反转单向链表和反转双向链表的函数

【要求】 如果链表长度为 N ，时间复杂度要求为 $O(N)$ ，额外空间复杂度要求为 $O(1)$

题目六

打印两个有序链表的公共部分

【题目】 给定两个有序链表的头指针head1和head2，打印两个链表的公共部分。

【要求】 如果两个链表的长度之和为N，时间复杂度要求为 $O(N)$ ，额外空间复杂度要求为 $O(1)$

面试时链表解题的方法论

- 1) 对于笔试，不用太在乎空间复杂度，一切为了时间复杂度
- 2) 对于面试，时间复杂度依然放在第一位，但是一定要找到空间最省的方法

重要技巧：

- 1) 额外数据结构记录（哈希表等）
- 2) 快慢指针

判断一个链表是否为回文结构

【题目】给定一个单链表的头节点head，请判断该链表是否为回文结构。

【例子】1->2->1，返回true； 1->2->2->1，返回true； 15->6->15，返回true；
1->2->3，返回false。

【例子】如果链表长度为N，时间复杂度达到 $O(N)$ ，额外空间复杂度达到 $O(1)$ 。

题目九

将单向链表按某值划分成左边小、中间相等、右边大的形式

【题目】 给定一个单链表的头节点head，节点的值类型是整型，再给定一个整数pivot。实现一个调整链表的函数，将链表调整为左部分都是值小于pivot的节点，中间部分都是值等于pivot的节点，右部分都是值大于pivot的节点。

【进阶】 在实现原问题功能的基础上增加如下的要求

【要求】 调整后所有小于pivot的节点之间的相对顺序和调整前一样

【要求】 调整后所有等于pivot的节点之间的相对顺序和调整前一样

【要求】 调整后所有大于pivot的节点之间的相对顺序和调整前一样

【要求】 时间复杂度请达到 $O(N)$ ，额外空间复杂度请达到 $O(1)$ 。

复制含有随机指针节点的链表

【题目】一种特殊的单链表节点类描述如下

```
class Node {  
    int value;  
    Node next;  
    Node rand;  
    Node(int val) {  
        value = val;  
    }  
}
```

rand指针是单链表节点结构中新增的指针，rand可能指向链表中的任意一个节点，也可能指向null。给定一个由Node节点类型组成的无环单链表的头节点head，请实现一个函数完成这个链表的复制，并返回复制的新链表的头节点。

【要求】时间复杂度 $O(N)$ ，额外空间复杂度 $O(1)$

两个单链表相交的一系列问题

【题目】 给定两个可能有环也可能无环的单链表，头节点head1和head2。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

【要求】 如果两个链表长度之和为N，时间复杂度请达到 $O(N)$ ，额外空间复杂度请达到 $O(1)$ 。

提升项目经验

- 课程名称：《牛客高级项目课--（牛客网）》
- 课程地址：<https://www.nowcoder.com/courses/semester/senior>
- 独家内部100元优惠券：DRMscjy



面试算法书籍

- 书名：《程序员代码面试指南—IT名企算法与数据结构题目最优解》
- 作者：左程云

THANK YOU

查看更多笔经面经

