# 预习周

chunk it up

deliberate practicing

feedback（主动feedback，看别人优秀的代码，被动feedback，同学或老师批改你的代码）

## 五遍刷题法：

1. 没想法直接看题解，并把所有可能的解法都照抄一遍，并理解
2. 默写一遍
3. 找到讨论区，点赞数最高的前几名的解法，并学习到别人好的思路和编码技术(前三名的代码)，并记录下来，自己模仿着运用
4. 一周以后再复习一遍
5. 面试前再复习一遍

## 切题四件套

1. 确定题目的前提条件
2. 找出所有可能的解题办法，并分析时间复杂度
3. 找到最优解后，开始写代码
4. 测试用例

## 指法和小操作

* fn+delete删除光标右边
* command+left/right 行头行尾
* option+left/right 光标按单词切分
* option+delete 删除单词
* shift+command+right 选中整行（光标在行头时）
* option+上下键（移动一整行）

## google code style

if 后面要有空格，左括号前面要有空格， 关键字和括号之前要有空格

## 自顶向下的编程方式

## 记忆知识点方法：画脑图

# 第一周

## 第三课 数组 链表 跳表

### 数组、链表、跳表的基本实现和特性

### 实战题目：移动零

移动数组下标的基本功

两层循环枚举数组的基本功

### 实战题目：盛水最多的容器，爬楼梯

盛水最多的容器

1. 暴力法，O(n^2),两层for循环求解最大面积时的左右柱子
2. 左右加逼方法，O(N), 比较两边较短的柱子然后往里逼近。

爬楼梯

**蒙逼的时候怎么办？**

**升维的思想，用空间换时间**

先从最简单的情况入手

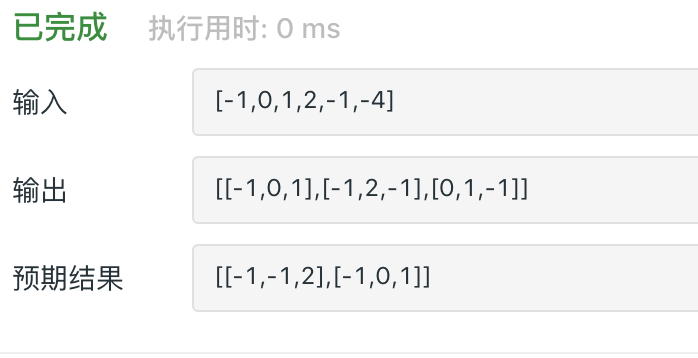
找最近的重复

### 实战题目解析：3数之和，环形链表

两数之和，a + b = teraget

三数之和等于0，a + b = - c

1. 暴力求解法 O(n^3),三层for循环，如何消除重复解的问题？



1. hash表（后续课程会讲）
2. 双指针左右下标，往中间推进，这种办法是很难想出来的，所以要记下来，做到举一反三，

一定要熟记这种双指针解法套路，并让自己的代码写的尽量简洁

环形链表

### Linked List 实战题目

解法固定，熟能生巧

主要在于自己多练习，思路很直接，代码一不小心会写得很复杂

课后作业

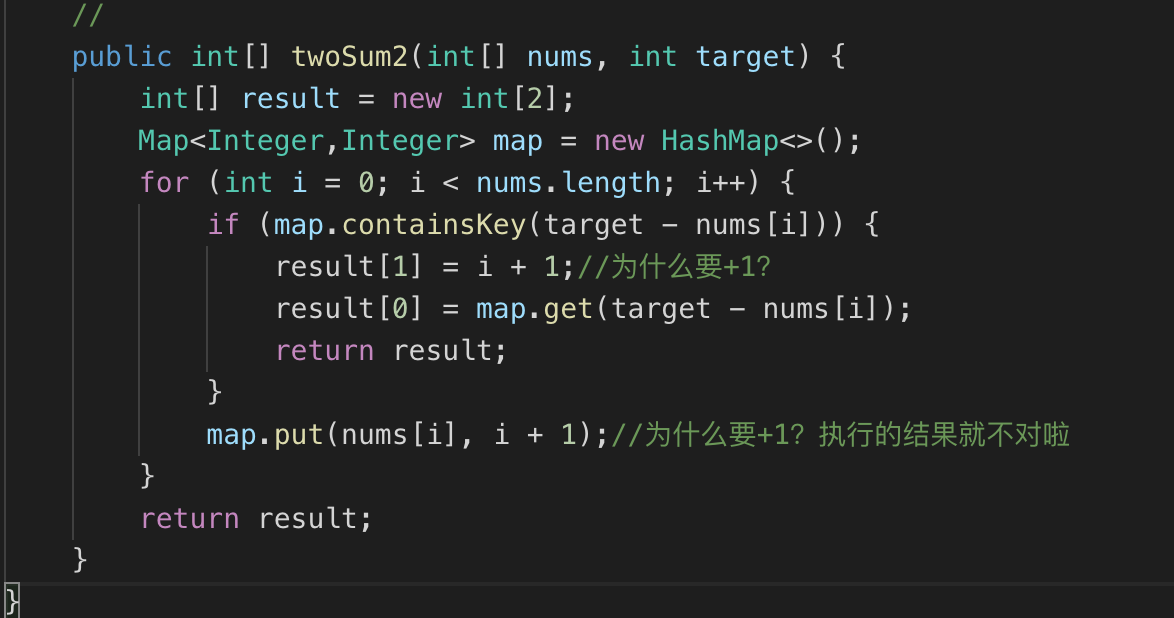
1. 两数之和

<https://leetcode-cn.com/problems/two-sum/>

1. 暴力法，on2，两层for 循环遍历匹配
2. 哈希表，O(n),把数组元素的值和位置用hashmap存储，遍历条件target-nums[i],看是否有key存在，则取位置值为输出结果。

记住这种用hash表存储值来判断的思路

疑问



## 第四课 栈、队列、优先队列、双端队列

### 栈和队列的实现和特性

栈stack：

先入后出；添加、删除皆为O(1)，查询o（n），因为元素是无序的需要遍历一遍。

Queue:

先入先出；添加、删除皆为O(1)，查询o（n）

实战中经常用的

双端队列Deque : Double- End Queue

1. 两端都可以进出的queue
2. 插入和删除都是o（1）操作，查询O(n)

Stack, Queue, Deque的工程实现

* Java, Python, C++ 等都有基础实现

以java为例，在bing或者google上搜索“stack java 10”就可以看到stack的官方文档，看它的api,直接推荐你使用deque。

掌握查询java类和接口文档的方法，stack，queue，deque在java api中的实现

作业：用新的api 改写左边的程序



add first remove last。。。

优先队列Priority Queue

1. 插入操作：O(1)
2. 取出操作：O(logn)变慢了-按元素的优先级取出（元素需要实现comparator）
3. 底层具体实现的数据结构较为多样和复杂：heap（二叉树堆，fibonlaci堆），bst， treap，
4. java 实现priority queue

作业：分析java queue和priority queue 是如何实现的？

从核心功能代码分析，异常分析，扩容策略，用什么数据结构实现的这几个角度来分析源码。

写在总结里面去

### 实战题目解析：有效的括号、最小栈等问题

找一个写md的好工具

# 第二周

## 第五课 哈希表、映射、集合

### 哈希表、映射、集合的实现与特性

### 哈希表

也叫散列表，是根据关键码值（key value）而直接进行访问的数据结构。

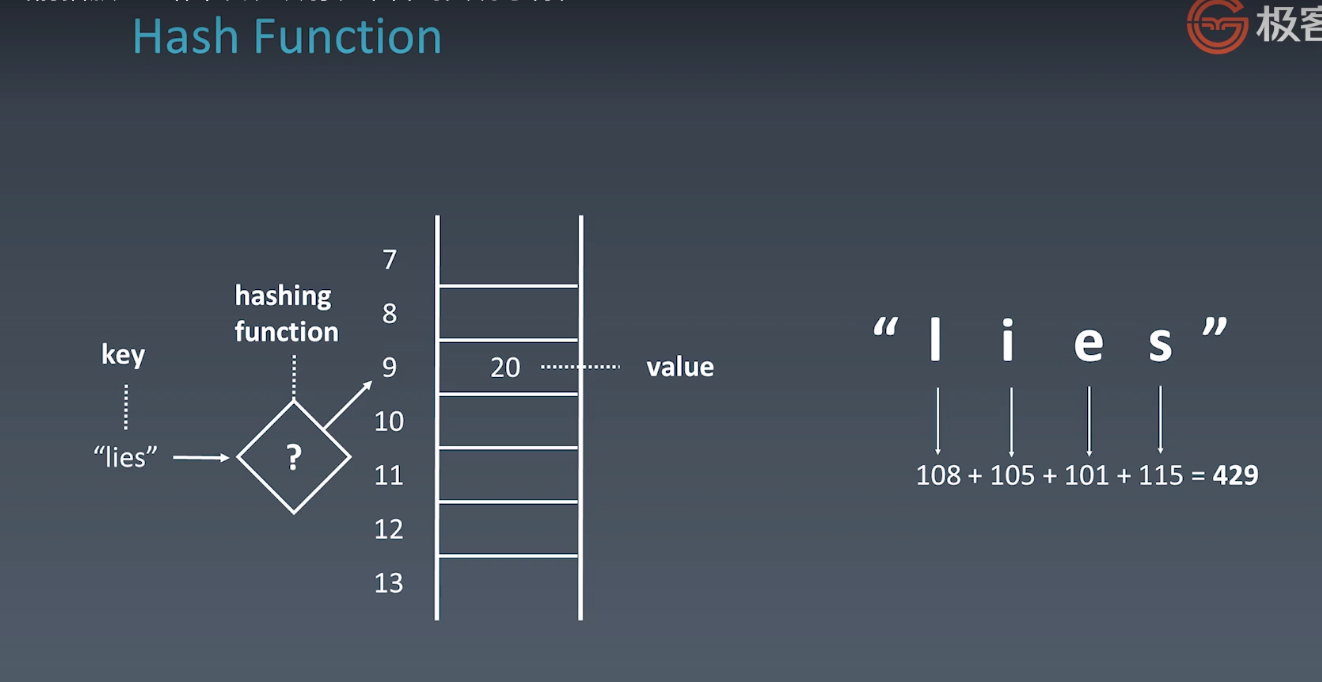
它通过把关键码值映射到表中一个位置来访问记录，以加快查找的速度。

这个映射函数叫散列函数（hash function），存放记录的数组叫作哈希表或散列表。

### 工程实践

电话号码薄，用户信息表，缓存(LRU Cache), 键值对存储(Redis)

### 原理



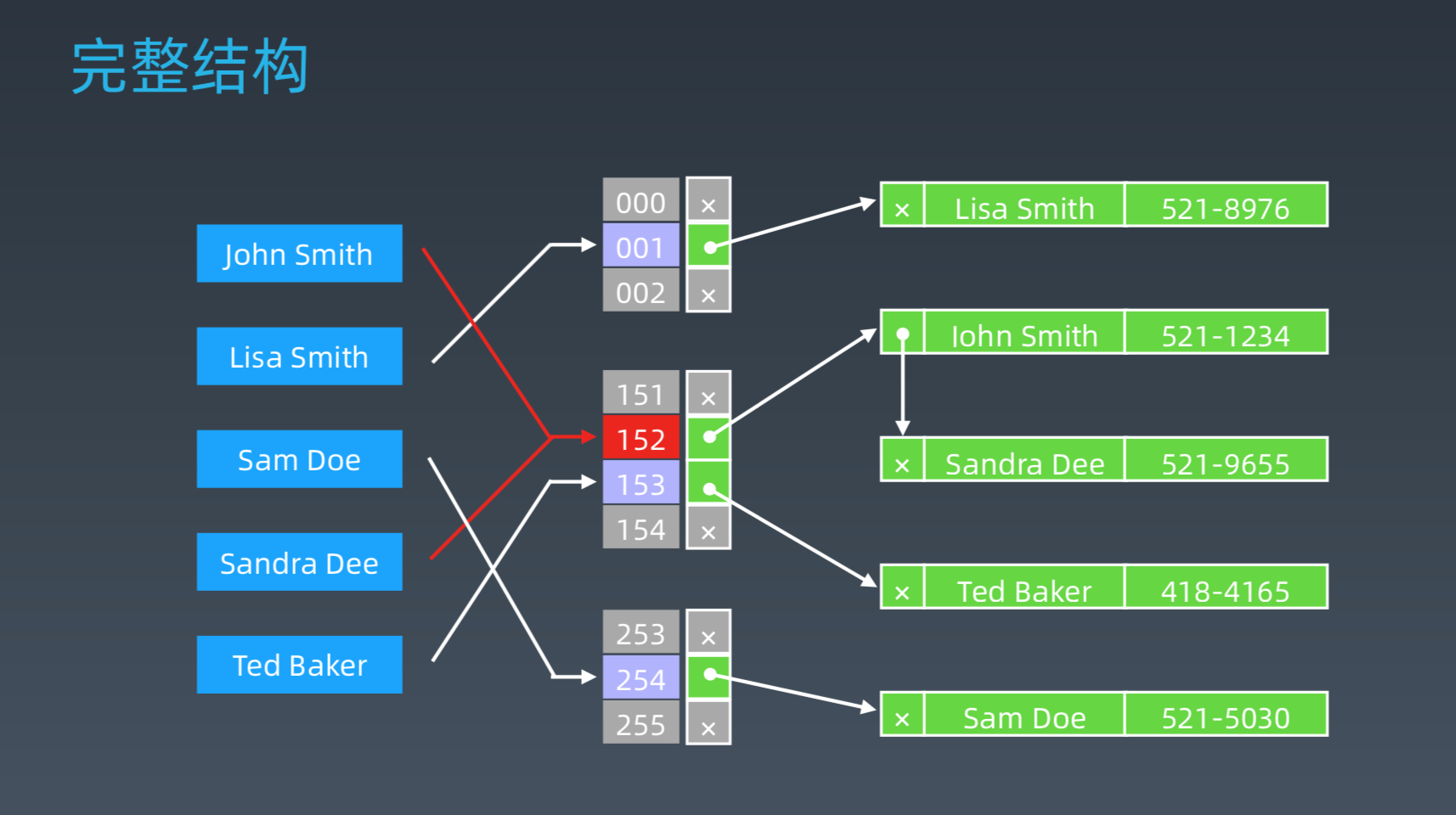
哈希碰撞



解决哈希冲突的办法：

在同一个位置拉一个链表，把都应该存储在这个位置的数据依次存储进来。

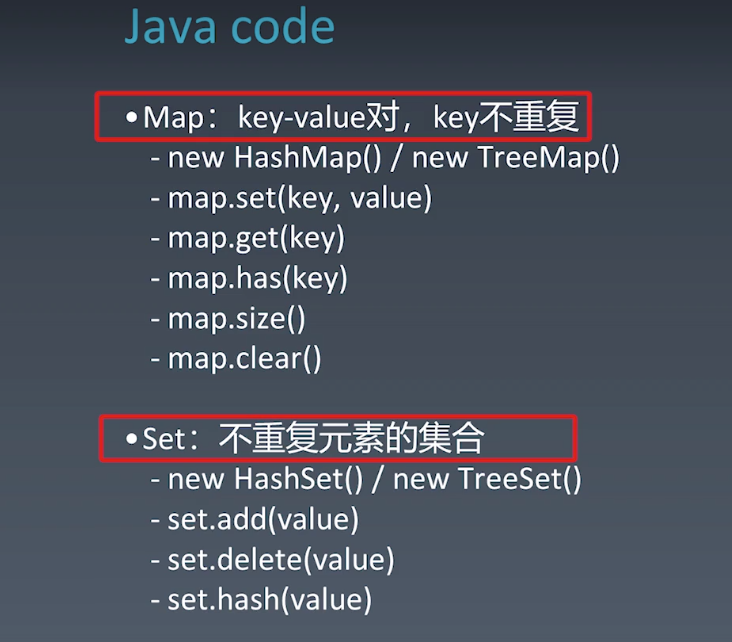
在哈希函数设计的较好的时候，碰撞较少的时候，添加、删除、查询是O(1)时间复杂度



### 复杂度分析

O(1),增加，删除，查询

### Java code



作业：看hashmap源码的put，putval和get函数就行了，用红黑树来实现，

所有复杂度都是logn

### 实战题目：有效的字母异位词等问题

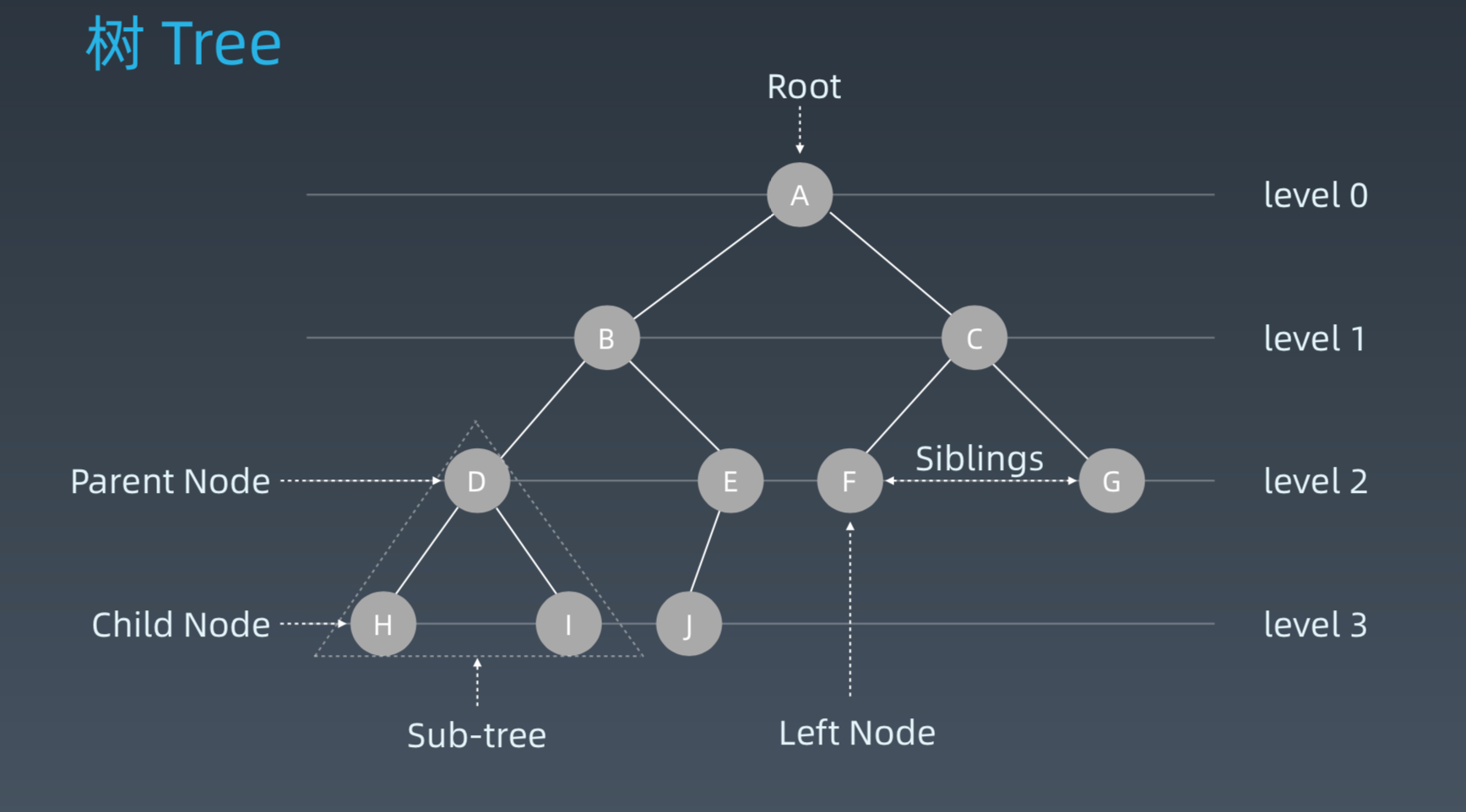
养成收藏精选代码的习惯

异位词分组第二种解法再研究一下

## 第六课 树、图、二叉树、二叉树搜索树

### 树、二叉树、二叉搜索树的实现和特性

想查询快最重要的思想：升维



s s s

### 实战题目解析：二叉树的中序遍历