

数据结构

时间复杂度与空间复杂度

- 时间复杂度，空间复杂度都有最好，平均，最坏情况
- 计算时间复杂度，空间复杂度的时候，不能只看代码。还要结合思想
- 时间复杂度，空间复杂度结合起来，衡量一个算法的好坏
- 计算时间复杂度，空间复杂度的方法：大O渐进法
- 常见的复杂度
  - $O(1) < O(\log N) < O(N) < O(N \cdot \log N) < O(N^2) < O(2^N)$

泛型

- 什么是泛型？--> 为了实现类型的参数化，表示该类里面只能放这个类类型的数据
- 不能直接new 一个泛型数组
- 泛型的编译
  - 泛型是编译时期的机制，在运行的时候没有泛型这个概念
  - 在编译的时候，所有的泛型占位符都会被擦除为Object
- 泛型的上界
  - 语法：class 泛型类名称<T extends A> {}，其中T的类型就只能是A和A的子类
  - 特殊的泛型上界，泛型的上界还可以是实现了某个接口，比如只要某类实现了这个接口那就是可以的
  - 注意没有泛型的下界这种说法
- 通配符
  - 通配符？
  - 通配符的上界 <? extends A> 表示只要是A和A的子类就可以匹配上
  - 通配符的下界 <? super A> 表示只要是A和A的父类就可以匹配上
- 包装类
  - 基本数据类型及其包装类—— 主要注意Integer，Character,其他全是首字母大写就是包装类
  - 装箱与拆箱
    - 装箱—— 将基本数据类型转换为其对应包装类
    - 拆箱—— 将包装类转换为其对应的基本数据类型
  - 注意，Integer类型的数据，值在[-128,127]之间的数据是在底层有数组缓存的，所以这个范围里面的数据的引用地址是一样的

ArrayList与顺序表

- 顺序表
  - 顺序表就是用一段连续的物理存储空间来存储元素的线性结构
  - 基本增删查改操作(需要我们自己实现)
- ArrayList
  - 是一个集合类，底层数据结构是一个动态增长的数据，该集合类集合了许多操作这个数组的方法，当然这是Java给我们提供好的
  - 了解其数组扩容机制，你直接赋值参数那么数组大小就是你赋值的值。如果没有赋值，那么刚开始数组大小就是0，在第一次add后变成10。扩容是按照1.5倍扩容
  - 底层源码

LinkedList与链表

- 链表—— 链表就是以一种非连续的物理存储结构，用节点来对数据进行组织
- 链表的基本操作，增删查改
- LinkedList
  - LinkedList是一个集合类，它底层的数据结构是一个双向链表，里面当然也含有很有操作链表的方法
  - LinkedList还可以当作栈，队列来使用，功能很强大
- 链表面试题：很重要，详情可以看看我的博客

栈和队列

- 栈
  - 是一种先进后出的数据结构
  - 可以实现为顺序的，也可以实现为链式的
  - 栈的应用
    - 最可能的出栈书顺序
    - 括号匹配问题
- 队列
  - 是一中先进先出的数据结构
  - 可以实现为顺序的，也可以实现为链式的
  - 双端队列，就是两端都可以进行入队，出队操作

结构很简单，但是题很常见，所以要多做题

二叉树

- 什么是树？树是一种非线性性的数据结构，它是一个由节点组织起来的具有层次的集合
- 判断是不是树的标准--> 子树之间不能有交集
- 树的表示形式
  - 双亲表示法
  - 孩子表示法
  - 孩子双亲表示法
  - 孩子兄弟表示法
- 满二叉树
- 完全二叉树
- 满二叉树和完全二叉树—— 满二叉树是一种特殊的完全二叉树，这二者的区分要会
- 节点数与层数之间的关系
  - 如果某棵树有k层，那么最大节点数就是 $2^k - 1$
  - 在第i层上，最大节点数  $2^{(i-1)}$
  - 如果一共有n个节点，那么层数为 $\log(n+1)$ 向上取整
- 二叉树的性质
  - 对于任何一棵二叉树，它的度为0的节点个数总是比度为2的多一个
  - 根节点从0开始编号，父亲节点下标为i，则左右孩子下标分别为  $2i + 1$  和  $2i + 2$ 。当然计算结果都不能超过n，否则就不存在左右孩子
  - 在完全二叉树里面，如果是偶数节点数，那么这棵树一定存在一个度为1的节点。如果奇数节点数，那么这棵树一定不存在度为1的节点
  - 由这个进而推出，总节点数G。如果G是偶数，那么叶子节点数等于G/2。如果G是奇数。那么就是  $(G+1) / 2$
- 二叉树的存储：顺序存储和类似于链表的链式存储。注意是类似于，不是链表
- 二叉树的操作：前中后序遍历

线性表

- 具有n个相同特性元素的有序序列
- 顺序表，链表，栈，队列都是属于线性表
- 线性表在逻辑上一定是连续的，但是在物理上不一定连续。线性结构存在顺序存储与链式存储两种物理存储方法

堆

- 堆是一棵完全二叉树。它的底层存储是顺序存储也就是数组。但是数组里面的元素顺序是按照完全二叉树的层序元素顺序来的
- 堆的种类
  - 大根堆
  - 小根堆—— 优先级队列底层默认建立的是小根堆。但是可以通过传入比较器修正为大根堆
- 堆的创建
  - 从最后一棵子树的根节点开始，一个个的进行向下调整
  - 建堆的时间复杂度 $O(n)$
- 堆的插入元素与删除元素
  - 插入：插入到数组末尾，开始进行向上调整
  - 删除：把堆顶元素与最后一个元素交换，UsedSize--，然后向下调整0下标这棵树
- 堆中元素都是可比较的元素。所以堆中不可以插入null,自定义类型需要实现Comparable接口或者传入比较器
- 堆的应用
  - topK问题
  - 第K大，第K小问题

排序

七大排序，时间复杂度，空间复杂度，稳定性，应用场景。详见博客

Map和Set

面试题：请你说一下ArrayList与LinkedList的区别

- 背后数据结构
  - ArrayList是一个动态数组
  - LinkedList是一个双向链表
- 存储方式
  - ArrayList是顺序存储结构，逻辑上连续，物理上也连续
  - LinkedList是链式存储结构，逻辑上连续，物理上不一定连续
- 空间利用机制
  - ArrayList底层是数组，元素个数达到一定标准就会进行扩容
  - LinkedList是链表，组织的是节点，有一个就链接一个，不存在扩容
- 各自优势
  - ArrayList底层是数组，所以是支持随机访问的，查询的效率比较高
  - LinkedList不支持随机访问，但是链表的插入和删除相较于ArrayList在一般情况下都是会更快的