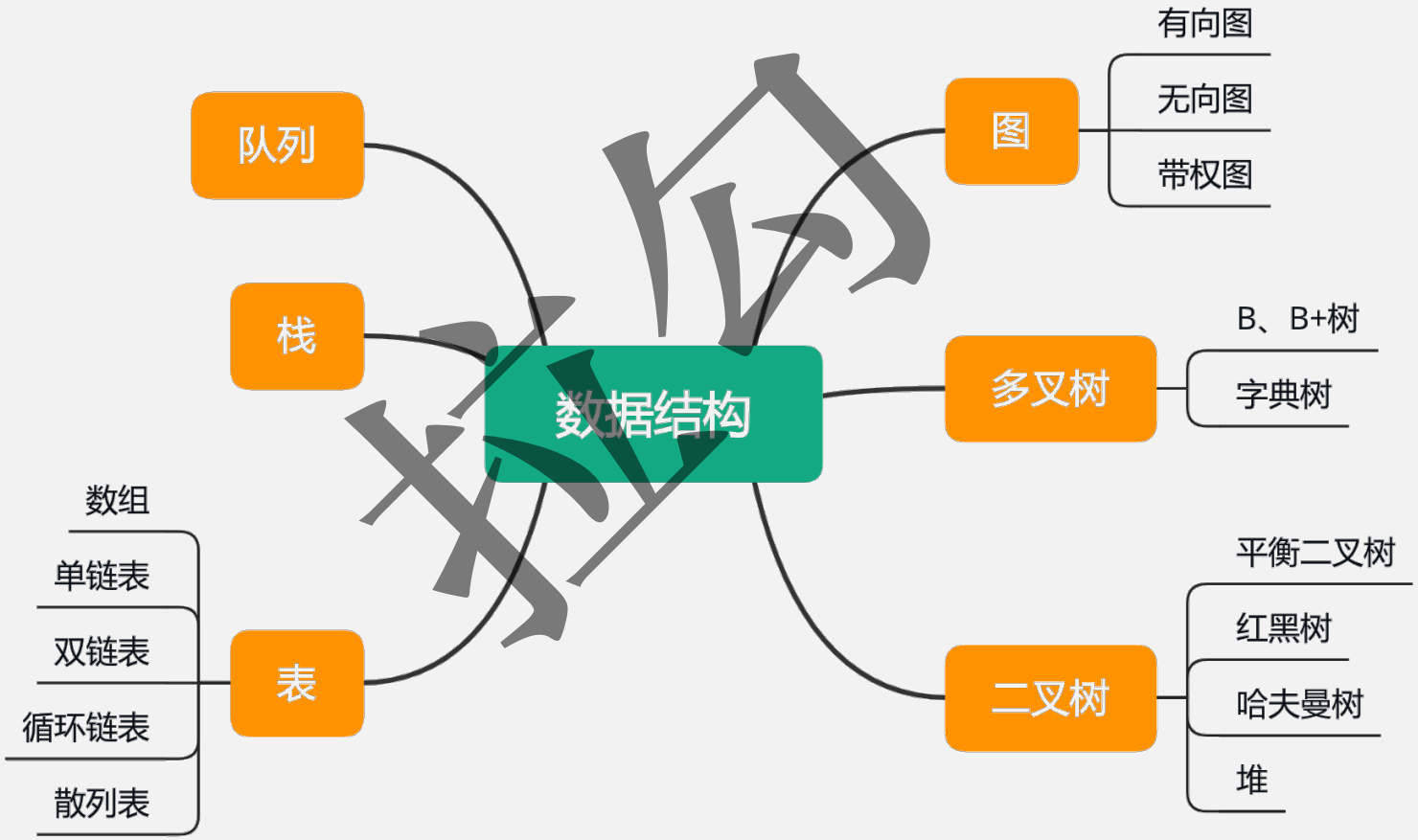
课时5

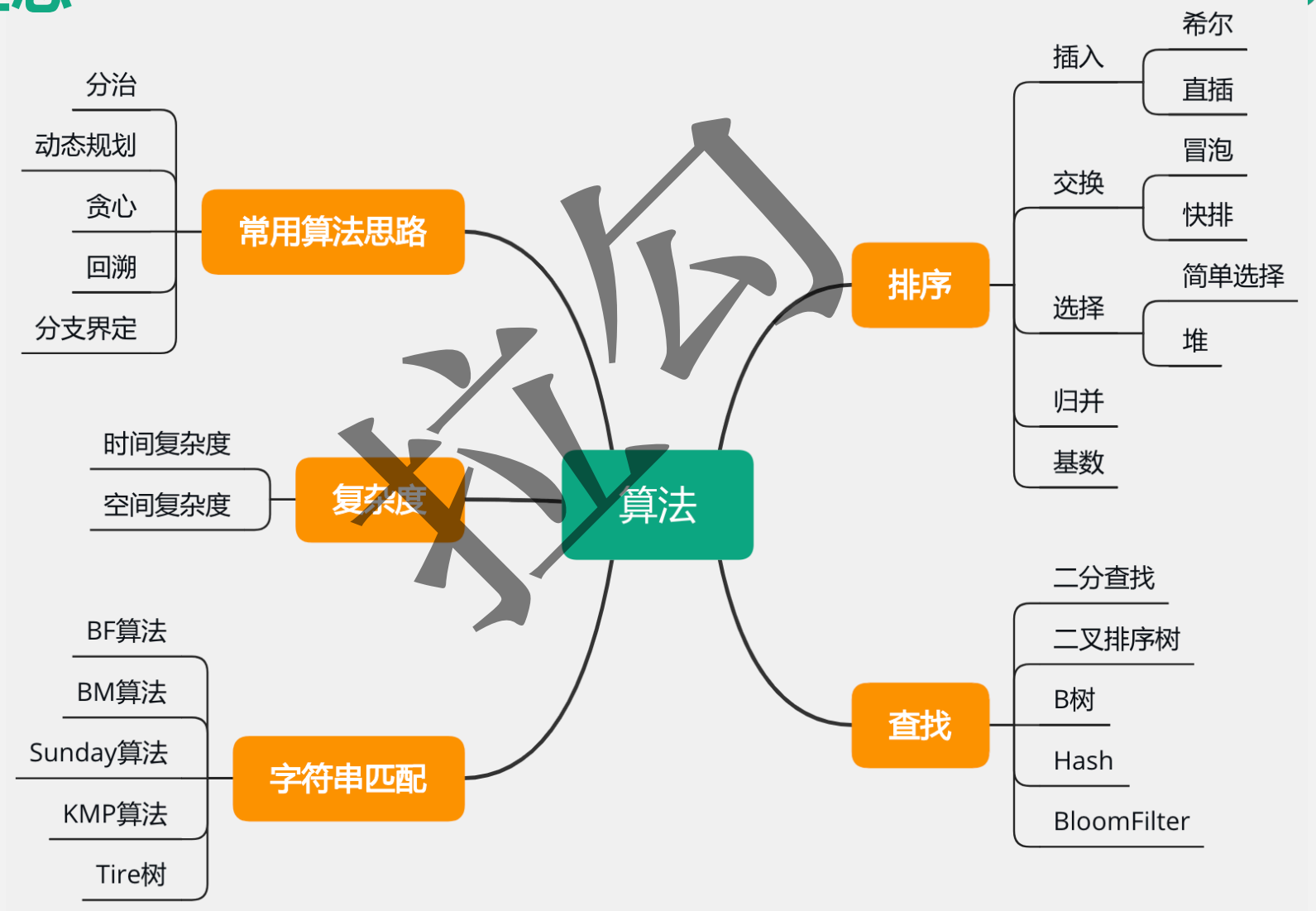
数据结构与算法

1. 知识点汇总
2. 从搜索树到B+树
3. 字符串匹配
4. TopK问题
5. 常用算法适用场景
6. 考察点和加分项
7. 真题

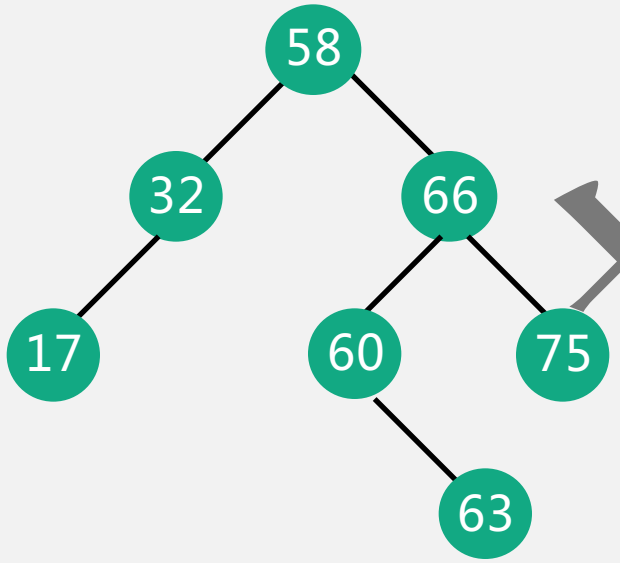
知识点汇总


数据结构
队列
栈
表
数组
单链表
双链表
循环链表
散列表
图
有向图
无向图
带权图
多叉树
B、B+树
字典树
二叉树
平衡二叉树
红黑树
哈夫曼树
堆

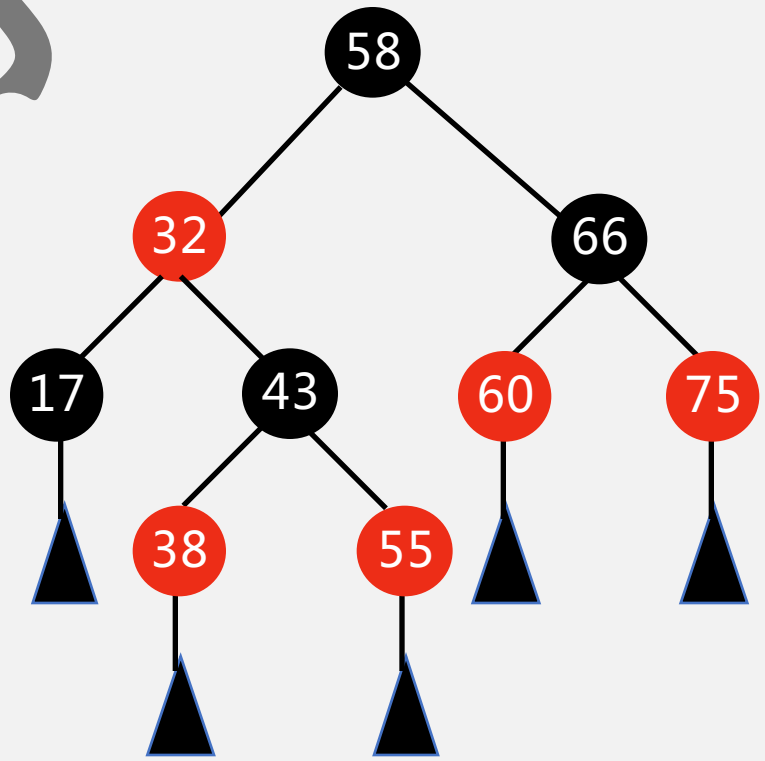
知识点汇总


算法
常用算法思路
分治
动态规划
贪心
回溯
分支界定
复杂度
时间复杂度
空间复杂度
字符串匹配
BF算法
BM算法
Sunday算法
KMP算法
Tire树
排序
插入
希尔
直插
交换
冒泡
快排
选择
简单选择
堆
归并
基数
查找
二分查找
二叉排序树
B树
Hash
BloomFilter

从搜索树到B+树



二叉搜索树

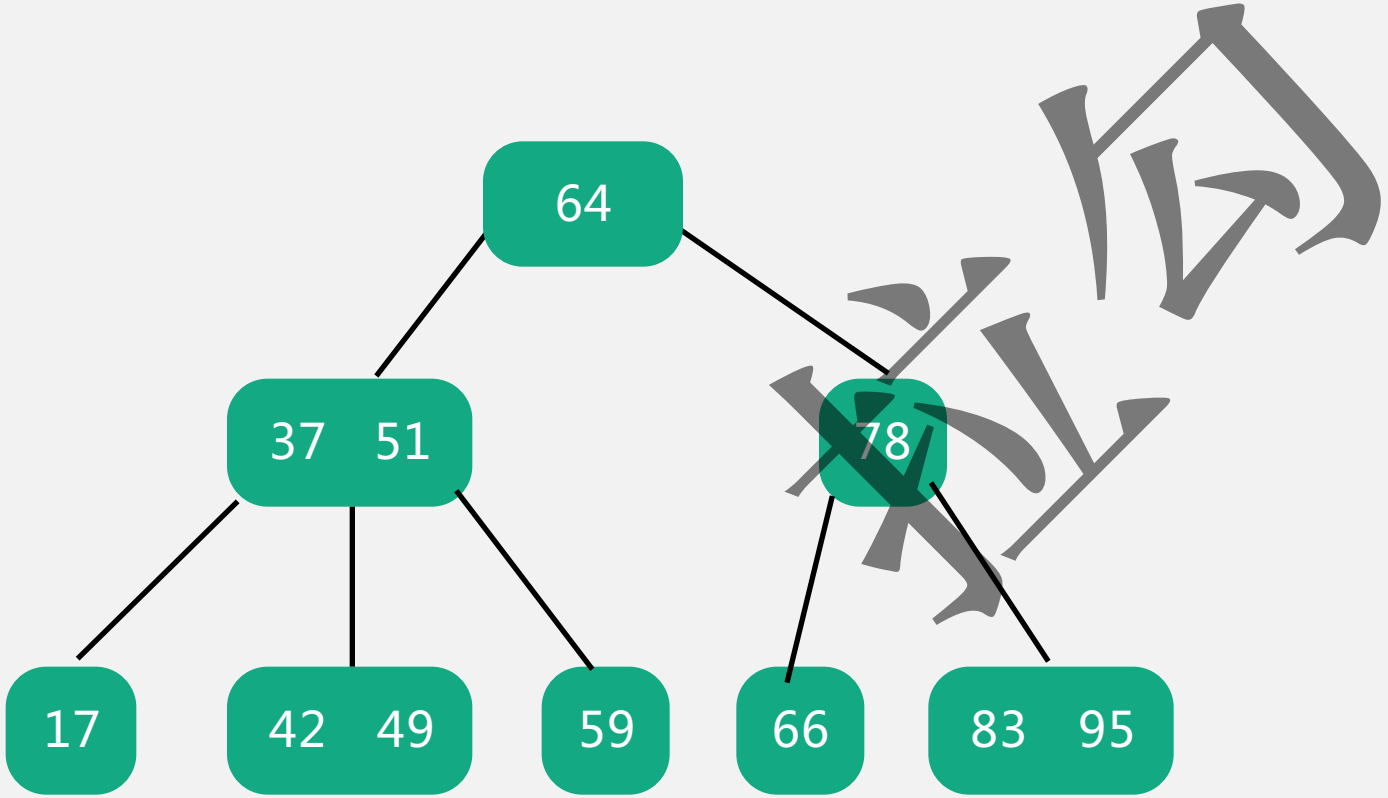


红黑树

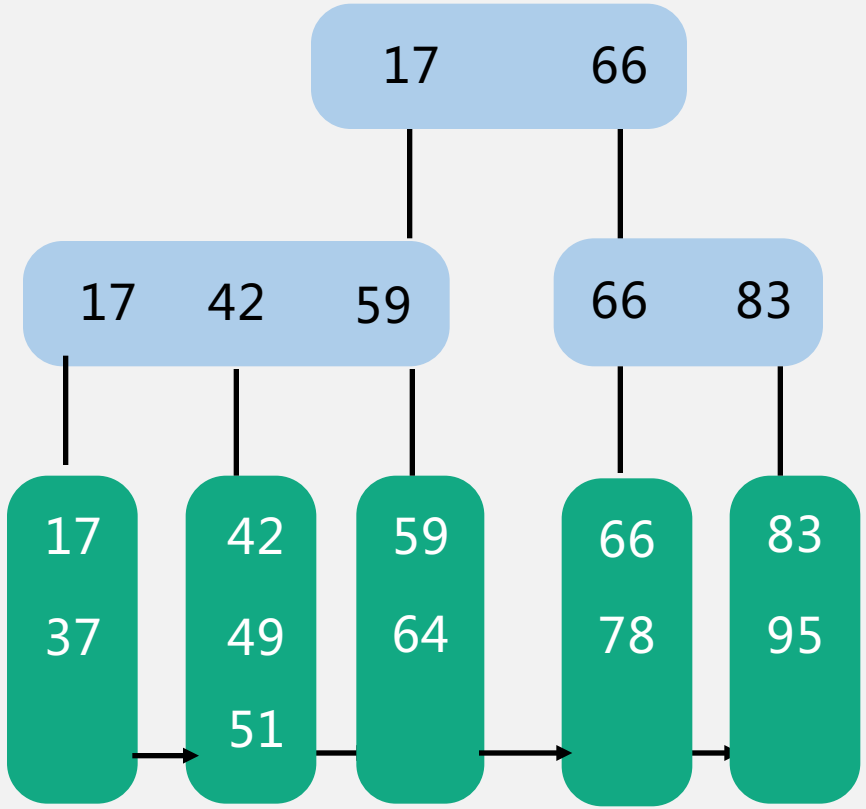
知识点详解

拉勾 LAGOU.COM x msup®

从搜索树到B+树



B-树



B+树

字符串匹配问题：

判断给定字符串中的括号是否匹配

解题思路：

1、使用栈

2、遇左括号入栈

3、遇右括号出栈，判断出栈括号是否与右括号成对

知识点详解


```
private static final Map<Character, Character> brackets = new HashMap<>();
static {
    brackets.put(')', '(');
    brackets.put(']', '[');
    brackets.put('}', '{');
}

public static boolean isMatch(String str) {
    if (str == null) {
        return false;
    }
    Stack<Character> stack = new Stack<>();
    for (char ch : str.toCharArray()) {
        if (brackets.containsValue(ch)) {
            stack.push(ch);
        } else if (brackets.containsKey(ch)) {
            if (stack.empty() || stack.pop() != brackets.get(ch)) {
                return false;
            }
        }
    }
    return stack.empty();
}
```

字符串匹配问题解题技巧

认真审题：

1、单模匹配 or 多模匹配

2、时间复杂度 or 空间大小

是否有要求

3、明确期望的返回值，例如

存在多个结果时如何处理

解题思路：

1、单模匹配：BM、KMP等

2、多模匹配：Tire树

3、前缀或后缀匹配

4、可以借助栈、树等数据结构

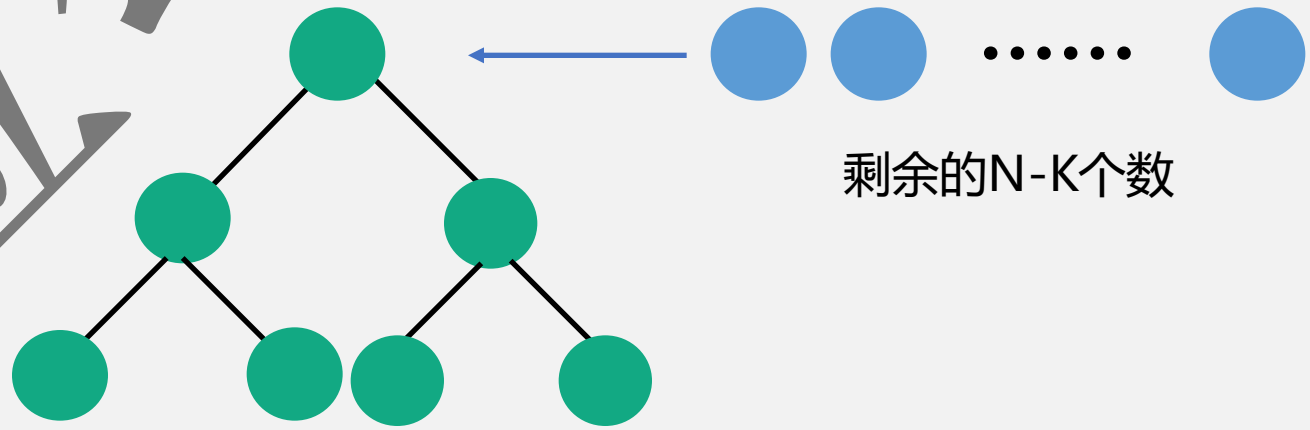
TopK问题：找出N个数中最小的K个数（N非常大）

解法：

1、用前K个数创建大小为K的大根堆

2、剩余N-K个数跟堆顶进行比较

时间复杂度：N*logK



TopK变种：从N有序队列中找到最小的K个值

解法：

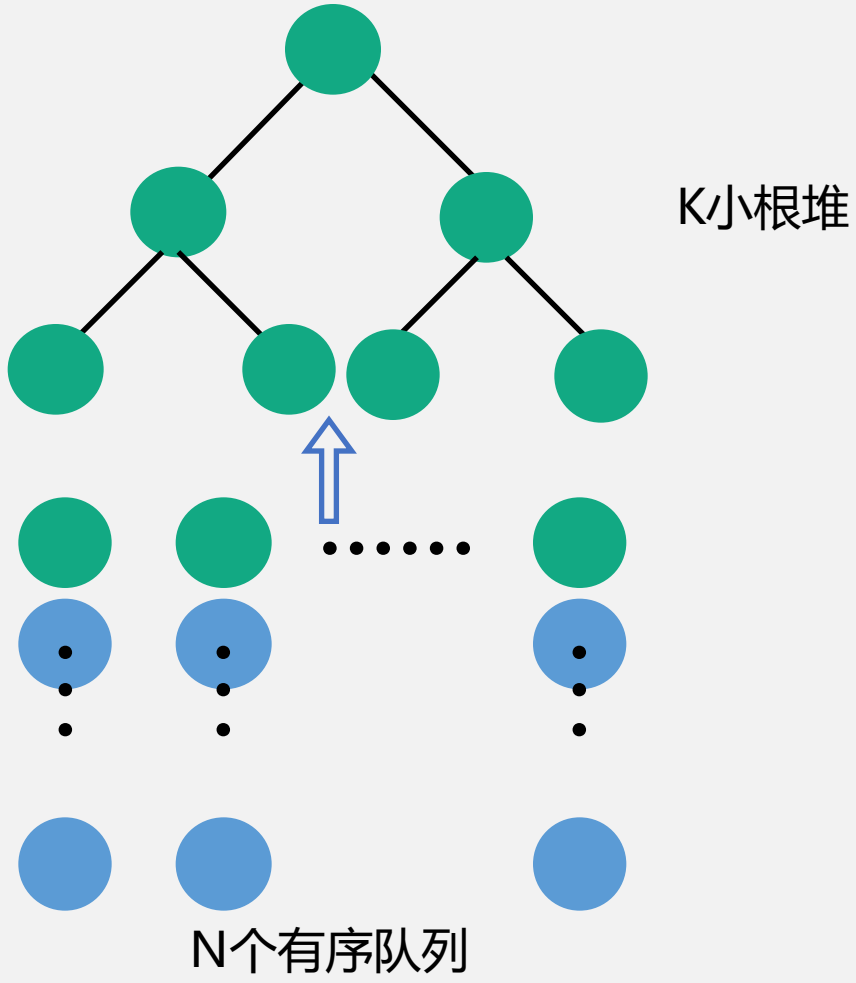
1、用N个队列的最小值组成大小为K的小根堆

2、取堆顶值

3、将堆顶值所在队列的下一个值加入堆，并调整

4、重复步骤2，直到K次

时间复杂度：(N+K-1)*logK



常用算法介绍

算法	适用场景	使用方法
分治	1、可以分解为子问题 2、子问题的解可以合并为原问题的解 3、子问题之间没有关联	1、找到最小子问题的求解方法 2、找到合并方法 3、找到递归终止条件
动态规划	1、按顺序求解子问题 2、子问题之间有关联关系 3、最后一个子问题的解为原问题的解	1、分析最优解的性质 2、递归的定义最优解 3、记录不同阶段的最优值 4、根据阶段最优解选择全局最优解

常用算法介绍

算法	适用场景	使用方法
贪心	1、局部最优解能产生全局最优解 2、具备后无效性	1、分解为子问题 2、计算每个子问题的局部最优解 3、合并局部最优解
回溯	1、深度优先搜索 2、获取解空间的所有解	1、定义解空间 2、确定是否进行扩展搜索的规则 3、深度优先遍历
分支界定	1、广度优先搜索 2、获取解空间的任意解	1、确定解特征 2、确定子节点搜索策略（FIFO、LIFO） 3、广度优先遍历

面试考察点

考察点

1. 了解基本数据结构及特点
2. 表、栈、队列、树需要熟练掌握，深刻理解使用场景
3. 了解常用的搜索、排序算法，及复杂度和稳定性
4. 了解常用的字符串处理算法
5. 能够分析算法实现的复杂度
6. 了解常用算法分类，解决问题的思路和解决哪类问题

面试加分项

拉勾 LAGOU.COM x msup®

加分项

1. 能够将数据结构与实际使用场景结合
2. 不同算法在业务场景中的应用
3. 面对模糊的题目能沟通确认条件和边界
4. 书写算法代码前，先讲一下解题思路
5. 能够发现解答中的一些问题，给出改进的思路

真题汇总-1

1. 各种排序算法实现和复杂度、稳定性
2. 二叉树的前、中、后序遍历
3. 翻转句子中单词的顺序
4. 用栈模拟队列（或用队列模拟栈）
5. 对10亿个数进行排序，限制内存为1G
6. 去掉（或找出）两个数组中重复的数字

真题汇总-2

7. 将一颗二叉树转换成其镜像
8. 确定一个字符串中的括号是否匹配
9. 给定一个开始词，一个结束词，一个字典，如何找到从开始词到结束词的最短单词接龙路径
10. 如何查找两个二叉树节点的最近公共祖先

关注订阅号：**IT进阶思维**，学习更多技术干货

Next：课时6《常用工具集》