数据结构知识框架

扫码回复【数四】
下载 520 场超清模版固定文件

红黑树

- 概念
  - 红黑树是一棵二叉搜索树，它在每个节点上增加了一个存储位来表示该点的颜色，颜色或是红色或是黑色的。通过对任一条从根到叶子的路径上颜色的限制，红黑树确保没有一条路径会比其他路径长出两倍，因而是接近平衡的。
  - 每个节点不是红色就是黑色
  - 根结点是黑色
  - 每个叶子节点（NIL 空节点）是黑色的
  - 每个红色节点的两个子结点都是黑色（也就是说，从叶子到根的所有路径上不能有两个连续的红色节点）
  - 对每个节点，从该节点到其叶子节点的简单路径上，均包含相同数目的黑色节点
  - 每个叶子节点都是黑色的（此处的叶子节点是指空子节点）
  - 红色节点，是从它到根的路径上的小于两条路径长度的节点的一半。换句话说，如果一条路径有 n 个节点，最长的路径长度不超过 2n，其中一半节点是黑色的，剩下的是红色的
  - 插入
  - 删除
  - 插入实现
  - 删除
  - 时间复杂度
  - 应用
  - 红黑树的Linux内核应用
- 结构
  - 定义
  - 性质
  - 旋转
  - 插入
  - 删除
  - 时间复杂度
  - 应用
  - 与2-3-4树的关系
  - 与AVL树的比较
  - 实现
  - 变体
  - 扩展阅读
- 实现
  - 数据结构
  - 算法
  - 工具
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

AVL树

- AVL树的定义
  - 平衡因子
  - 左子树
  - 右子树
  - 左旋转
  - 右旋转
  - 左右旋转
  - 右左旋转
  - 插入
  - 删除
  - 查找
  - 时间复杂度
  - 空间复杂度
  - 平衡维护
  - 应用场景
  - 与红黑树的比较
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

二叉搜索树

- 定义
  - 二叉搜索树（BST）的定义
  - 节点结构
  - 左子树
  - 右子树
  - 父节点
  - 键值
  - 唯一性
  - 无重复键
  - 处理重复键
  - 性质
  - 中序遍历
  - 有序性
  - 子树性质
  - 递归性
  - 操作
  - 插入
  - 删除
  - 查找
  - 最小值
  - 最大值
  - 前驱
  - 后继
  - 时间复杂度
  - 最坏情况
  - 平均情况
  - 空间复杂度
  - 递归实现
  - 迭代实现
  - 内存管理
  - 应用
  - 数据库索引
  - 文件系统
  - 编译器
  - 与其他数据结构的比较
  - 与哈希表
  - 与平衡二叉树
  - 与堆
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

二叉堆

- 概念
  - 二叉堆的定义
  - 完全二叉树
  - 堆性质
  - 最大堆
  - 最小堆
  - 父节点
  - 左子节点
  - 右子节点
  - 数组表示
  - 索引计算
  - 父节点索引
  - 左子节点索引
  - 右子节点索引
  - 操作
  - 插入
  - 删除
  - 查找
  - 最大值
  - 最小值
  - 时间复杂度
  - 插入
  - 删除
  - 查找
  - 最大值
  - 最小值
  - 空间复杂度
  - 内存布局
  - 动态扩容
  - 应用
  - 优先级队列
  - 堆排序
  - 图算法
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

哈希表

- 概念
  - 哈希表的定义
  - 哈希函数
  - 键值对
  - 桶
  - 冲突
  - 开放寻址法
  - 线性探测
  - 二次探测
  - 双哈希
  - 拉链法
  - 单链表
  - 动态数组
  - 红黑树
  - 哈希函数的设计
  - 均匀性
  - 确定性
  - 高效性
  - 抗碰撞性
  - 常见哈希函数
  - 除法哈希
  - 乘法哈希
  - 全域哈希
  - 字符串哈希
  - 加密哈希
  - 负载因子
  - 定义
  - 阈值
  - 动态扩容
  - 重新哈希
  - 时间复杂度
  - 平均情况
  - 最坏情况
  - 空间复杂度
  - 内存布局
  - 开放寻址法的内存
  - 拉链法的内存
  - 缓存友好性
  - 应用
  - 数据库索引
  - 缓存系统
  - 编译器符号表
  - 与其他数据结构的比较
  - 与二叉搜索树
  - 与堆
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

图

- 概念
  - 图的定义
  - 顶点
  - 边
  - 有向图
  - 无向图
  - 加权图
  - 无权图
  - 连通性
  - 连通图
  - 强连通图
  - 弱连通图
  - 表示方法
  - 邻接矩阵
  - 邻接表
  - 关联矩阵
  - 遍历
  - 深度优先搜索（DFS）
  - 递归实现
  - 迭代实现
  - 广度优先搜索（BFS）
  - 应用场景
  - 最短路径
  - 单源最短路径
  - 多源最短路径
  - 最小生成树
  - 拓扑排序
  - 强连通分量
  - 关键路径
  - 最大流
  - 最小割
  - 二分图匹配
  - 欧拉路径
  - 哈密顿路径
  - 旅行商问题（TSP）
  - 时间复杂度
  - 空间复杂度
  - 内存布局
  - 缓存友好性
  - 与其他数据结构的比较
  - 与树
  - 与哈希表
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

算法

- 分类
  - 按设计思想
  - 分治
  - 动态规划
  - 贪心
  - 回溯
  - 枝剪
  - 按时间复杂度
  - 常数时间 O(1)
  - 对数时间 O(log n)
  - 线性时间 O(n)
  - 线性对数时间 O(n log n)
  - 多项式时间 O(n^k)
  - 指数时间 O(2^n)
  - 阶乘时间 O(n!)
  - 按空间复杂度
  - 原地算法（In-place）
  - 辅助空间 O(1)
  - 辅助空间 O(n)
  - 递归深度
  - 按稳定性
  - 稳定算法
  - 不稳定算法
  - 稳定性的重要性
  - 按确定性
  - 确定性算法
  - 随机算法
  - 伪随机算法
  - 蒙特卡洛算法
  - 拉斯维加斯算法
  - 按应用领域
  - 排序
  - 搜索
  - 图论
  - 字符串
  - 数学
  - 密码学
  - 机器学习
  - 分布式
  - 并行
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

排序

- 概念
  - 排序问题的定义
  - 输入
  - 输出
  - 稳定性
  - 稳定排序的定义
  - 稳定性的重要性
  - 内存
  - 原地排序
  - 非原地排序
  - 时间复杂度
  - 最坏情况
  - 平均情况
  - 最好情况
  - 空间复杂度
  - 辅助空间
  - 递归栈
  - 比较排序
  - 基于比较的排序
  - 决策树模型
  - 下界证明
  - 非比较排序
  - 计数排序
  - 基数排序
  - 桶排序
  - 适用条件
  - 整数排序
  - 字符串排序
  - 浮点数排序
  - 外部排序
  - 多路归并
  - 磁盘I/O
  - 内存层次结构
  - 缓存友好性
  - 并行排序
  - 分治策略
  - 归并排序
  - 快速排序
  - 堆排序
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

搜索

- 概念
  - 搜索问题的定义
  - 输入
  - 输出
  - 目标
  - 成功搜索
  - 失败搜索
  - 搜索空间
  - 状态空间
  - 搜索图
  - 盲搜
  - 广度优先搜索（BFS）
  - 深度优先搜索（DFS）
  - 迭代加深搜索（IDS）
  - 双向搜索
  - 启发式搜索
  - 贪心最佳优先搜索
  - A* 搜索
  - 启发函数
  - 可接受性
  - 一致性
  - 最短路径问题
  - 单源最短路径
  - 多源最短路径
  - 广义最短路径
  - 最小生成树
  - 拓扑排序
  - 强连通分量
  - 关键路径
  - 最大流
  - 最小割
  - 二分图匹配
  - 匈牙利算法
  - 欧拉路径
  - 哈密顿路径
  - 旅行商问题（TSP）
  - 时间复杂度
  - 最坏情况
  - 平均情况
  - 最好情况
  - 空间复杂度
  - 辅助空间
  - 递归栈
  - 内存布局
  - 缓存友好性
  - 并行搜索
  - 分布式搜索
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

算法设计

- 概念
  - 算法设计的定义
  - 问题
  - 输入
  - 输出
  - 约束
  - 目标函数
  - 最优解
  - 可行解
  - 近似解
  - 启发式解
  - 设计目标
  - 正确性
  - 时间复杂度
  - 空间复杂度
  - 可读性
  - 可维护性
  - 可扩展性
  - 可测试性
  - 可复用性
  - 鲁棒性
  - 容错性
  - 安全性
  - 设计原则
  - 分治
  - 动态规划
  - 贪心
  - 回溯
  - 枝剪
  - 启发式搜索
  - 随机化
  - 近似
  - 设计模式
  - 递归
  - 迭代
  - 缓存
  - 并行
  - 分布式
  - 流式
  - 在线
  - 离线
  - 实现
  - 测试
  - 性能
  - 文档
  - 部署
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 未来趋势
  - 挑战
  - 总结

算法实现

- 实现方式
  - 从零实现
  - 使用标准库
  - 使用第三方库
  - 参数化设计
  - 模板
  - 泛型
  - 宏
  - 特化
  - 性能优化
  - 内存管理
  - 缓存友好性
  - 并行化
  - 向量化
  - 内联
  - 编译期计算
  - 测试
  - 单元测试
  - 集成测试
  - 属性测试
  - 模糊测试
  - 基准测试
  - 回归测试
  - 覆盖率
  - 测试数据
  - 测试夹具
  - 断言
  - 模拟
  - 桩
  - 文档
  - 接口文档
  - 实现文档
  - 设计文档
  - 教程
  - 示例
  - 常见问题
  - 迁移指南
  - 部署
  - 发布
  - 版本管理
  - 兼容性
  - 向后兼容
  - 向前兼容
  - 破坏性变更
  - 弃用
  - 迁移
  - 维护
  - 扩展
  - 学习资源
  - 社区
  - 贡献
  - 代码审查
  - 拉取请求
  - 问题跟踪
  - 讨论
  - 沟通
  - 文化
  - 未来趋势
  - 挑战
  - 总结