

算法分析与设计复习

带*为重重点

算法分析

合并查找

基于比较的排序*

查找算法

符号 二叉查找树

图算法*

最小生成树

字符串*

算法分析

观察法

线性回归不考

评估 $T(N)$ 的参数

技巧：两边取

倍率实验

规模翻倍

运行时间

$$T(2N) / T(N) = a(2N)^b / a(N^b) = 2^b = \text{ratio}$$

$$b = \lg(\text{ratio})$$

数学模型进行时间复杂度分析

精准模型

开销模型

取一种操作进行分析

应取最为频繁的操作meg

消除了cosi

波浪线标记

取高阶项

考试的时候 系数要保留

增长量级

$\lg N$

$N, N \log N$

N^2

2^N

$N!$

$(N! > 2^N)$

算法理论

最好、最坏、平均情况

最优算法

空间性能分析

精准分析：所有空间

只对使用较大的eg.数组

合并查找

连通性问题及抽象

三种算法

复习点

1. 数据结构 第*i*个元素表示是什么
2. 算法过程
3. 时间性能
4. 路径压缩的原理

weightedQuickUnion

算法的改进策略

1. 分析问题
2. 提出思路
3. 实现思路
4. 空间换时间算法设计思想

eg.weightedQuickUnion

基于比较的排序*

分治算法思想

归并排序（自顶自底都要会）

1. 过程
2. 分析
3. 改进

排序算法的性质（所学算法的比较（复习下朱讲的那个表））

原地

稳定性

复杂度

最优的排序算法

快速排序

算法过程：重点在划分

最好最坏情况

平均情况下的分析方法：计算数学期望

算法改进：引入插入排序，采样中位数（尽可能接近中位数）

快速选择算法：划分的应用

三向划分的场景

优先队列

初级实现

二叉堆

概念

上浮

贤臣

与初级实现性能方面的比较

堆排序:先入再出

查找算法

概念

初级实现

有序操作函数

符号 二叉查找树

概念

基本操作 插入删除

实现有序操作

删除操作与实现 lazy Hibbard

图算法*

MST

概念

割

谭颖思想

三种算法（把课上表复习） 算法思想 割 数据结构 实现 性能

索引优先队列

SP

SP性质 边的松弛 SP求解策略

Dijkstra*

不同算法 使用场景 策略 数据结构 实现

字符串*

要掌握

计数排序

LSD

PLUS：算法设计的建议(考试无关)

优先考虑经典的算法思想

分治 动态规划 贪心算法

搜索的方法

尽可能小的求解空间

设计剪枝函数

优化技术

空间换时间：查找表

设计合适的数据结构

程序性能优化

推荐本书《深入理解计算机系统》

高性能计算

并行计算 OpenMP

分布式计算 音译 SPK HDP

近似算法（不能保证精确解）