

需要复习的知识点

第1章 绪论（重点：数据结构概念）

第2章 线性表

第3章 栈和队列

第4章 串（一般了解）

第5章 数组和广义表（一般了解）

第6章 树和二叉树

第7章 图

第9章 查找

第10章 内部排序

带*的内容除外

第1章 绪论

数据结构的基本概念和术语

数据结构基本概念：逻辑结构、物理结构

数据结构的4种逻辑结构：指集合中元素之间的关系

数据结构的存储结构，算法的实现取决于物理结构

物理结构又体现在静态结构和动态结构

第1章 绪论

算法的描述和分析

算法、算法的重要特性(5个)、算法设计的要求(4个)

衡量算法的好坏：算法的时间复杂度和空间复杂度

算法的时间复杂度不仅仅依赖于问题的规模，也取决于输入实例的初始状态

算法描述和算法分析的方法，对于一般算法能分析出时间复杂度

第2章

线性表

线性表的逻辑结构

线性表的逻辑结构特征

线性表上定义的基本运算，并能利用基本运算构造出较复杂的运算

线性表的顺序存储结构

顺序表的含义及特点

顺序表上的插入（移动结点的平均次数： $n/2$ ）、删除操作（移动结点的平均次数 $(n-1)/2$ ）及其平均时间复杂度分析（都为 $O(n)$ ）

利用顺序表设计算法解决简单的应用问题

线性表的链式存储结构

链表中头指针和头结点的使用

单链表、双链表、循环链表链接方式上的区别

单链表上实现的查找（按序号、按值）、插入和删除（注意修改指针的顺序）等基本算法，并分析其时间复杂度（都为 $O(n)$ ）。

循环链表上尾指针取代头指针的作用，以及单循环链表上的算法与单链表上相应算法的异同点。

双链表的定义特点，求前驱较方便

利用链表设计算法解决简单的应用问题：如两个链表的合并等

顺序表和链表的比较

顺序表和链表的主要优缺点（从空间和时间这两方面来考虑，其基本操作实现的难易程度）

顺序表：随机存取结构，适宜于静态查找

链表：适宜于动态的插入、删除操作

针对线性表上所需要执行的主要操作，知道选择顺序表还是链表作为其存储结构才能取得较优的时空性能。

第3章

栈和队列

栈的逻辑结构、存储结构及其相关算法

栈的逻辑结构特点、栈与线性表的异同

顺序栈和链栈上实现的进栈、退栈等基本算法

栈的“上溢”、“下溢”的概念及判别条件

队列的逻辑结构、存储结构及其相关算法

队列的逻辑结构特点、队列与线性表的异同

顺序队列（主要是循环队列）和链队列上实现的入队、出队等基本算法

循环队列中对边界条件的处理方法

栈和队列的应用

栈和队列的特点，什么样的情况下能够使用栈和队列

第4章

串

串及其运算

串与线性表的关系：特殊的线性表

串的有关术语：空格串、空串、子串的位置、串相等、主串、子串等

串的基本操作：串赋值、串比较、求串长、串联接、求子串

串的模式匹配

第5章

数组和广义表

数组是一种随机存取结构

数组的两种表示方法：以列序为主序、以行序为主序。

特殊矩阵的概念和压缩存储

稀疏矩阵的概念和稀疏矩阵的三元组表表示方法

广义表的概念

第6章 树和二叉树

树的概念

树的常用术语：结点的度、树的度、树的深度、路径、路径长度等

二叉树概念

完全二叉树、满二叉树

一般用二叉链表存储

若二叉树有 n 个结点，则共有 $2n$ 个指针域，用 $n-1$ 个指针域指向孩子结点，则 $n+1$ 个指针域为空。

顺序存储结构适用于完全二叉树

二叉树的性质

要非常熟悉五个性质

三个性质对任意二叉树都适用：每层结点数至多为 2^{i-1} 、 $n_0 = n_2 + 1$ 、深度为k的二叉树的结点数最多为 $2^k - 1$

第四个性质对n个结点的完全二叉树而言，其深度为 $\lfloor \log_2 n \rfloor + 1$

第五个性质：左右孩子的编号

二叉树的存储结构

顺序存储结构（数组）

链式存储结构（二叉链表）

静态二叉链表和静态三叉链表

二叉树的遍历

常用三种遍历：先序、中序、后序

给定一棵二叉树的先序、中序和后序遍历序列是唯一的，反之不行

用先序和中序两个序列能唯一确定一棵二叉树

用中序和后序两个序列能唯一确定一棵二叉树

线索二叉树（重点：中序）

求指定结点的前驱和后继

画出线索树

遍历

树和森林

树的存储结构及特点：多重链表、双亲表示法、孩子链表、孩子兄弟表示法

树的遍历：先根、后根

树、森林、二叉树的转换

赫夫曼树及其应用

最优二叉树、前缀编码定义

构建赫夫曼树：度为1的结点数为0

赫夫曼编码

第7章

图

图的基本术语

邻接点、入度、出度、简单路径、连通和连通图、连通分量等

图的存储结构的特点及适用范围

邻接矩阵

邻接表

图的基本运算

图的遍历

深度优先遍历

广度优先遍历

与树的遍历之间的关系

生成树和最小生成树

生成树和最小生成树的概念

深度优先生成树和广度优先生成树

最小生成树

- Prim算法（适用于稠密图，时间复杂度为 $O(n^2)$ ）
- Kruskal算法（适用于稀疏图，时间复杂度为 $O(e \log_2 e)$ ）

关键路径：针对有向图AOE网

拓扑排序：针对有向图AOV网

拓扑排序的基本思想和步骤

最短路径

第9章

查找

查找的基本概念

查找结构

查找的判定树

平均查找长度

静态查找

顺序表：顺序查找算法、分析

有序表：折半查找算法、分析

动态查找

◆ 二叉排序树（二叉查找树）

- 定义

- 查找、平均查找长度

- 构造、插入、删除

◆ 平衡二叉树（AVL树）

- 定义

- AVL树的构建过程

◆ B-树

- 定义

- 查找思想

■ 静态查找结构

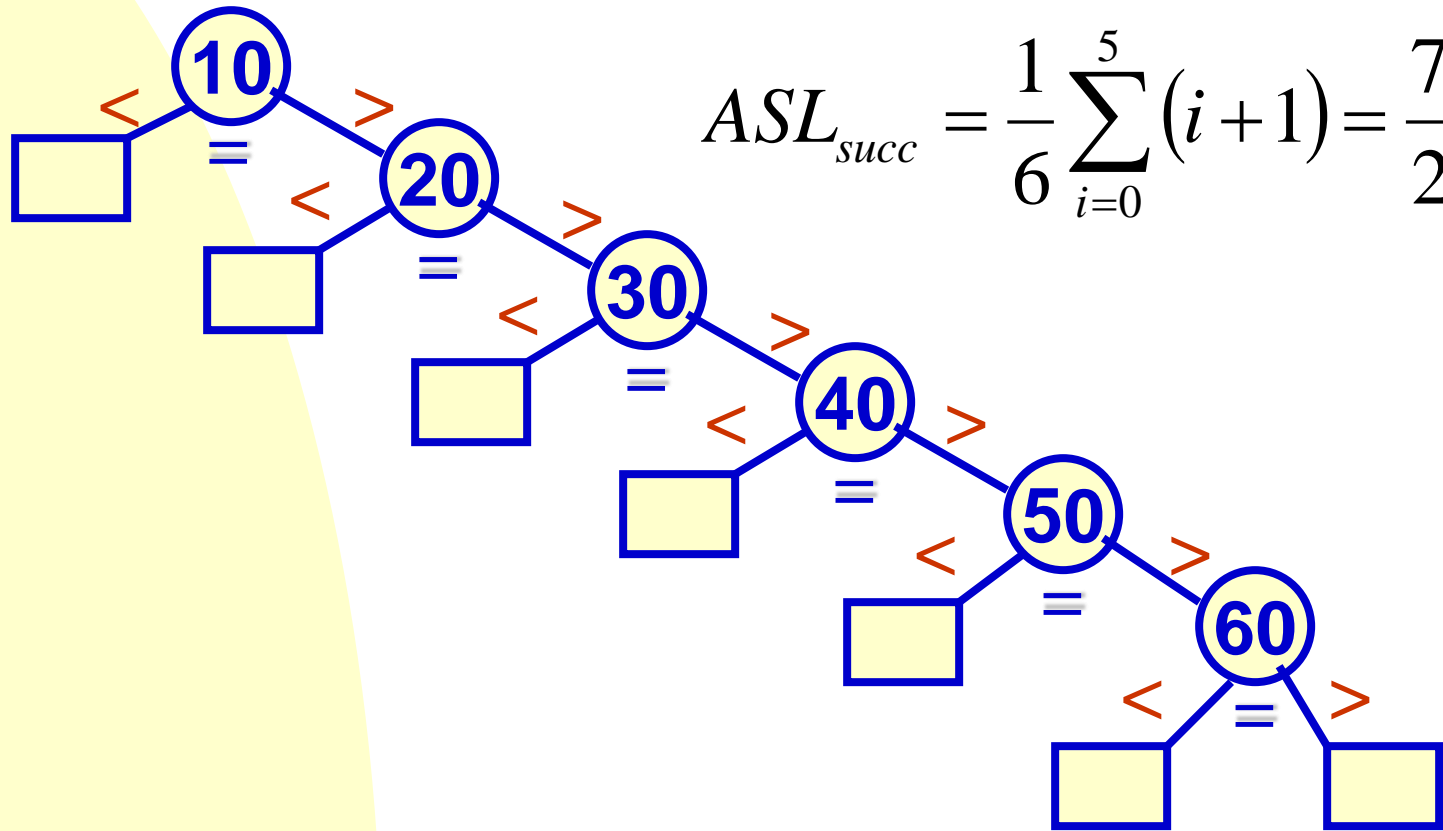
- ◆ 顺序查找 — 顺序表、链表
- ◆ 折半查找 — 有序顺序表

■ 动态查找结构

- ◆ 二叉排序树 — 无重复关键字
- ◆ *AVL*树 — 平衡二叉排序树

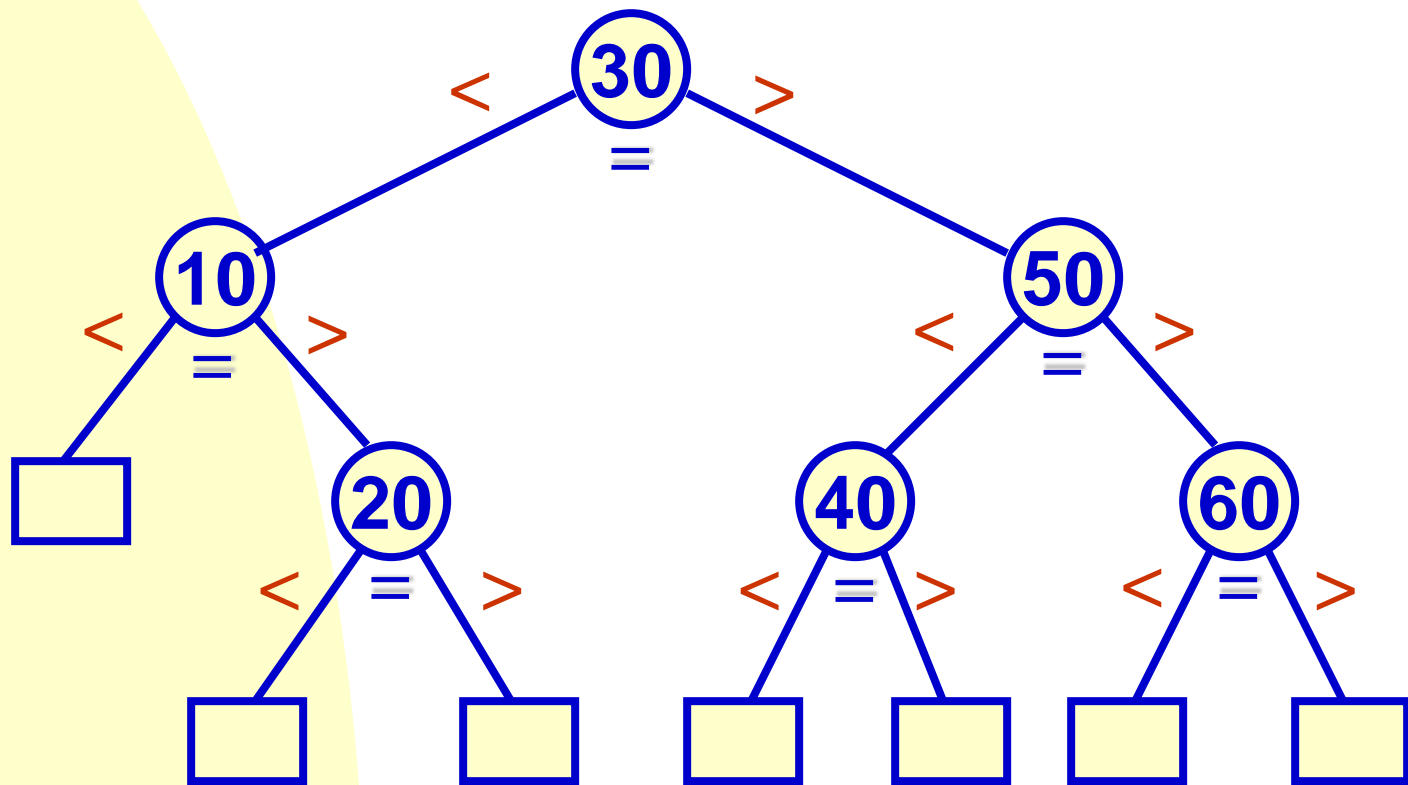
■ 有序顺序表的顺序查找

(10, 20, 30, 40, 50, 60)



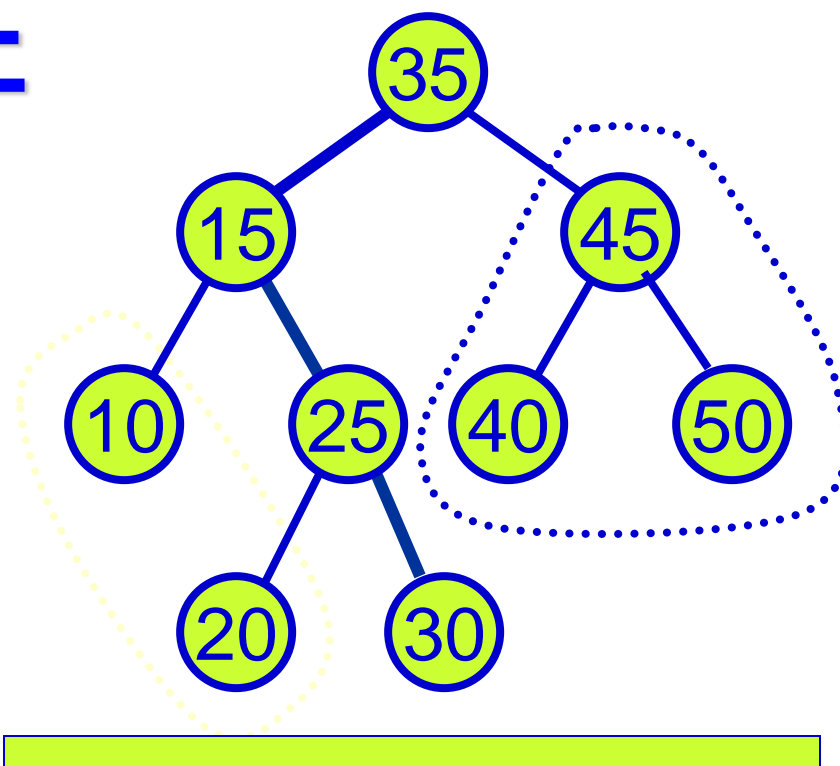
■ 有序顺序表的折半查找

(10, 20, 30, 40, 50, 60)



■ 二叉查找树

- ◆ 二叉查找树的子树是二叉查找树
- ◆ 结点左子树上所有关键字小于结点关键字
- ◆ 右子树上所有关键字大于结点关键字

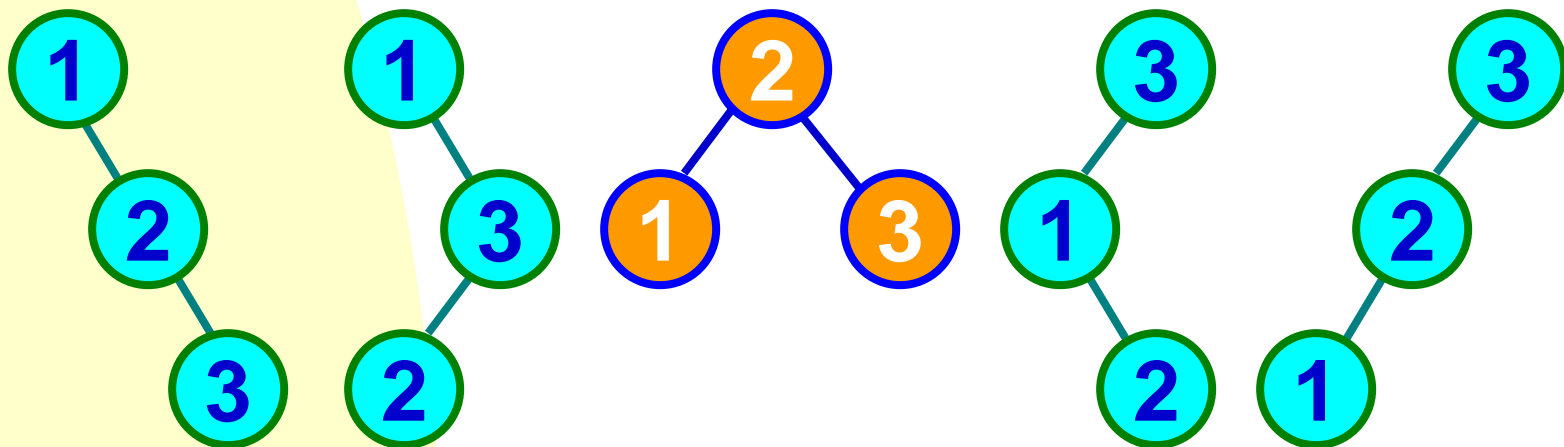


n 个结点的二叉查找树的数目

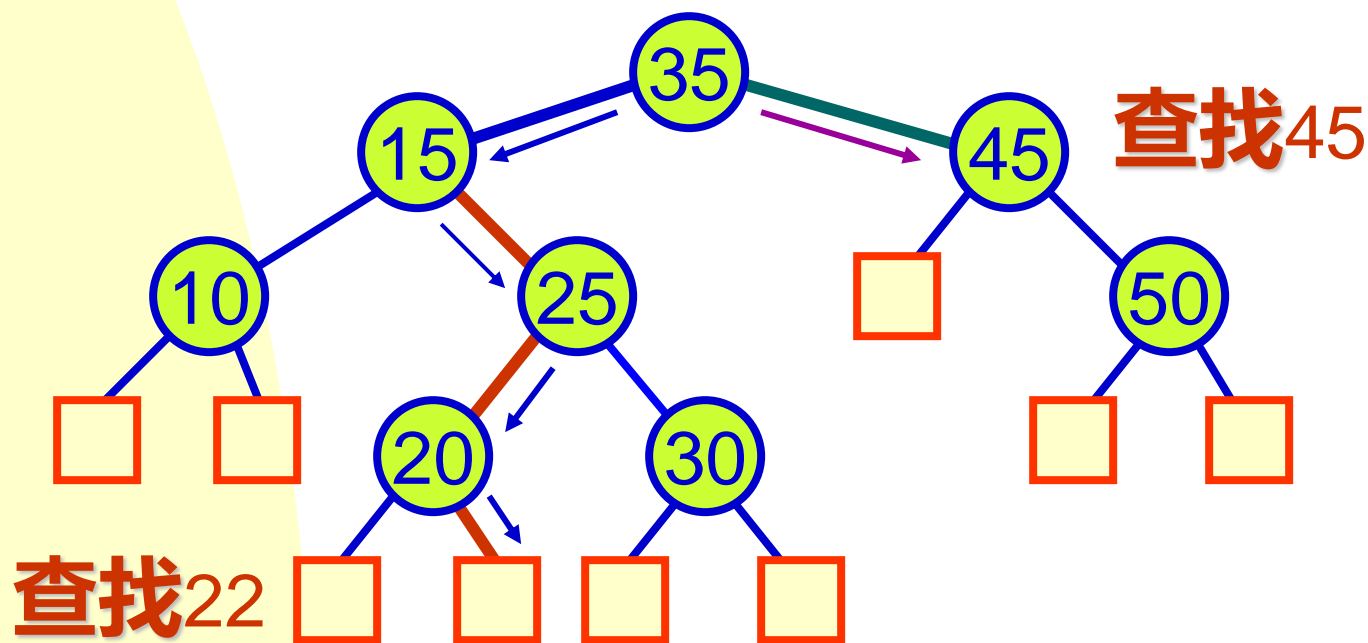
【例】3 个结点的二叉查找树

$$\frac{1}{3+1} C_{2*3}^3 = \frac{1}{4} * \frac{6*5*4}{3*2*1} = 5$$

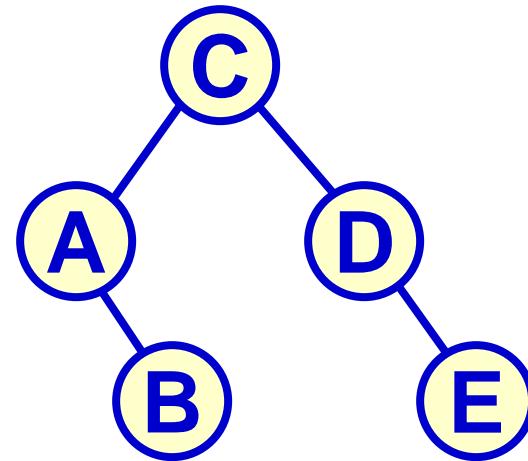
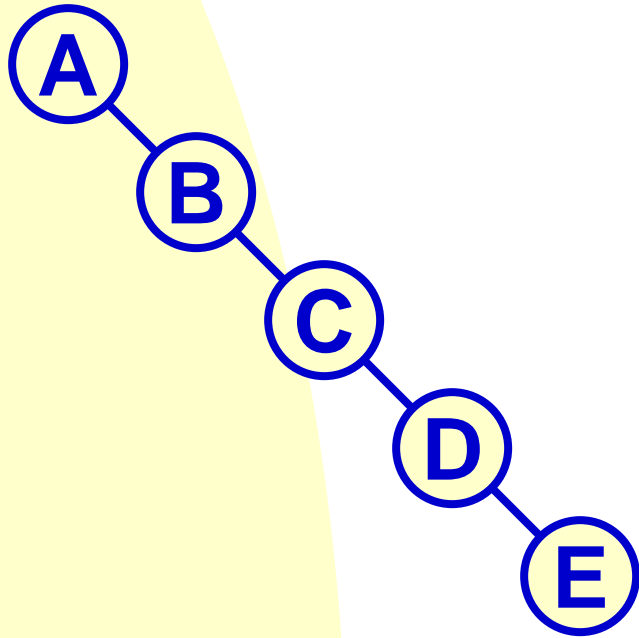
{123} {132} {213} {231} {312} {321}



- ◆ 查找成功时检测指针停留在树中某个结点。
- ◆ 查找不成功时检测指针停留在某个外结点（失败结点）。



- 二叉查找树的高度越小，平均查找长度越小。
- n 个结点的二叉查找树的高度最大为 $n-1$ ，最小为 $\lfloor \log_2 n \rfloor$ 。



■ *AVL*树

- ◆ 理解： *AVL*树的子树也是*AVL*树
- ◆ 掌握： 插入新结点后平衡化旋转的方法

散列技术、哈希表

有关概念：散列表、散列函数、散列地址等

散列函数的选取原则（计算哈希函数所需时间；关键字的长度；哈希表长度（哈希地址范围）；关键字分布情况；记录的查找频率。）

几种常用的散列函数构造方法：直接定址法、数字分析法、平方取中法、折叠法、除留余数法

解决冲突的方法：开放定址法（散列地址的计算公式是： $H_i(\text{key}) = (H(\text{key}) + d_i) \text{ MOD } m, i=1, 2, \dots, k(k \leq m-1)$ ）、再哈希法、链地址法等

散列表的构造

第10章 排序

- 排序的基本概念
 - ◆ 排序的基本概念
 - ◆ 关键字、初始关键字排列
 - ◆ 关键字比较次数、数据移动次数
 - ◆ 稳定性
 - ◆ 附加存储
- 几种排序方法的基本思想和特点
 - ◆ 插入排序、交换排序、选择排序、归并排序

■ 插入排序

- ◆ 用事例表明直接插入排序、折半插入排序的过程
 - ◆ 直接插入排序和折半插入排序的算法
 - ◆ 排序的性能分析
- 当待排序的关键字序列已经基本有序时，用直接插入排序最快

■ 交换排序

- ◆ 用事例表明起泡排序和快速排序的过程
- ◆ 起泡排序的算法
- ◆ 快速排序的算法
- ◆ 性能分析
- 起泡排序可实现部分排序
- 快速排序是一个递归的排序法
- 当待排序关键字序列已经基本有序时，快速排序显著变慢。

■ 选择排序

- ◆ 用事例表明直接选择排序、锦标赛排序、堆排序的过程
 - ◆ 直接选择排序和堆排序的算法
 - ◆ 三种选择排序的性能分析
- 用直接选择排序在一个待排序区间中选出最小的数据时，与区间第一个数据对调，不是顺次后移。这导致方法不稳定。

- 当在 n 个数据 (n 很大) 中选出最小的 5 ~ 8 个数据时, **锦标赛排序** 最快。
- 锦标赛排序算法将待排序数据个数 n 补足到 2 的 k 次幂
$$2^{k-1} < n \leq 2^k$$
- 在堆排序中将待排序的数据组织成完全二叉树的顺序存储。(大顶堆、小顶堆)

■ 二路归并排序

- ◆ 用事例表明二路归并排序的过程
- ◆ 二路归并排序的算法
- ◆ 该算法的性能分析
- 归并排序可以递归执行
- 归并排序需要较多的附加存储
- 归并排序对待排序关键字的初始排列不敏感，故排序速度较稳定。

各种排序方法的比较

| 排 序 方 法 | 比较次数 | | 移动次数 | | 稳 定 性 | 附加存储 | |
|---------|--------------|-------|--------------|-------|-------|------------|-------|
| | 最好 | 最差 | 最好 | 最差 | | 最好 | 最差 |
| 直接插入排序 | n | n^2 | 0 | n^2 | √ | 1 | |
| 折半插入排序 | $n \log_2 n$ | | 0 | n^2 | √ | 1 | |
| 起泡排序 | n | n^2 | 0 | n^2 | √ | 1 | |
| 快速排序 | $n \log_2 n$ | n^2 | $n \log_2 n$ | n^2 | × | $\log_2 n$ | n^2 |
| 简单选择排序 | n^2 | | 0 | n | × | 1 | |
| 锦标赛排序 | $n \log_2 n$ | | $n \log_2 n$ | | √ | n | |
| 堆排序 | $n \log_2 n$ | | $n \log_2 n$ | | × | 1 | |
| 归并排序 | $n \log_2 n$ | | $n \log_2 n$ | | √ | n | |

考核内容

- 1、基本知识、概念（40%）
- 2、综合应用（40%）
- 3、算法理解与设计（20%）

题型

- 1、选择题
- 2、填空或判断题
- 3、综合题
- 4、算法阅读题
- 5、算法设计题