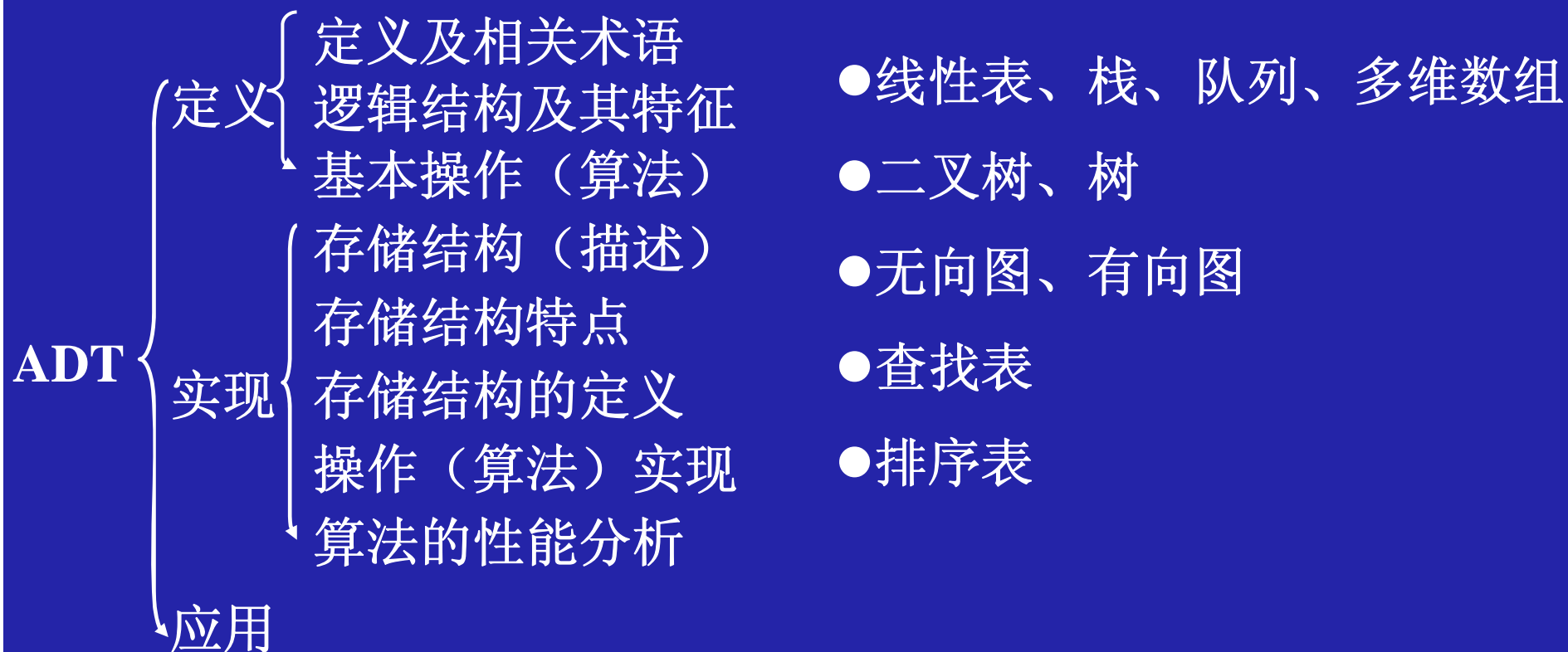
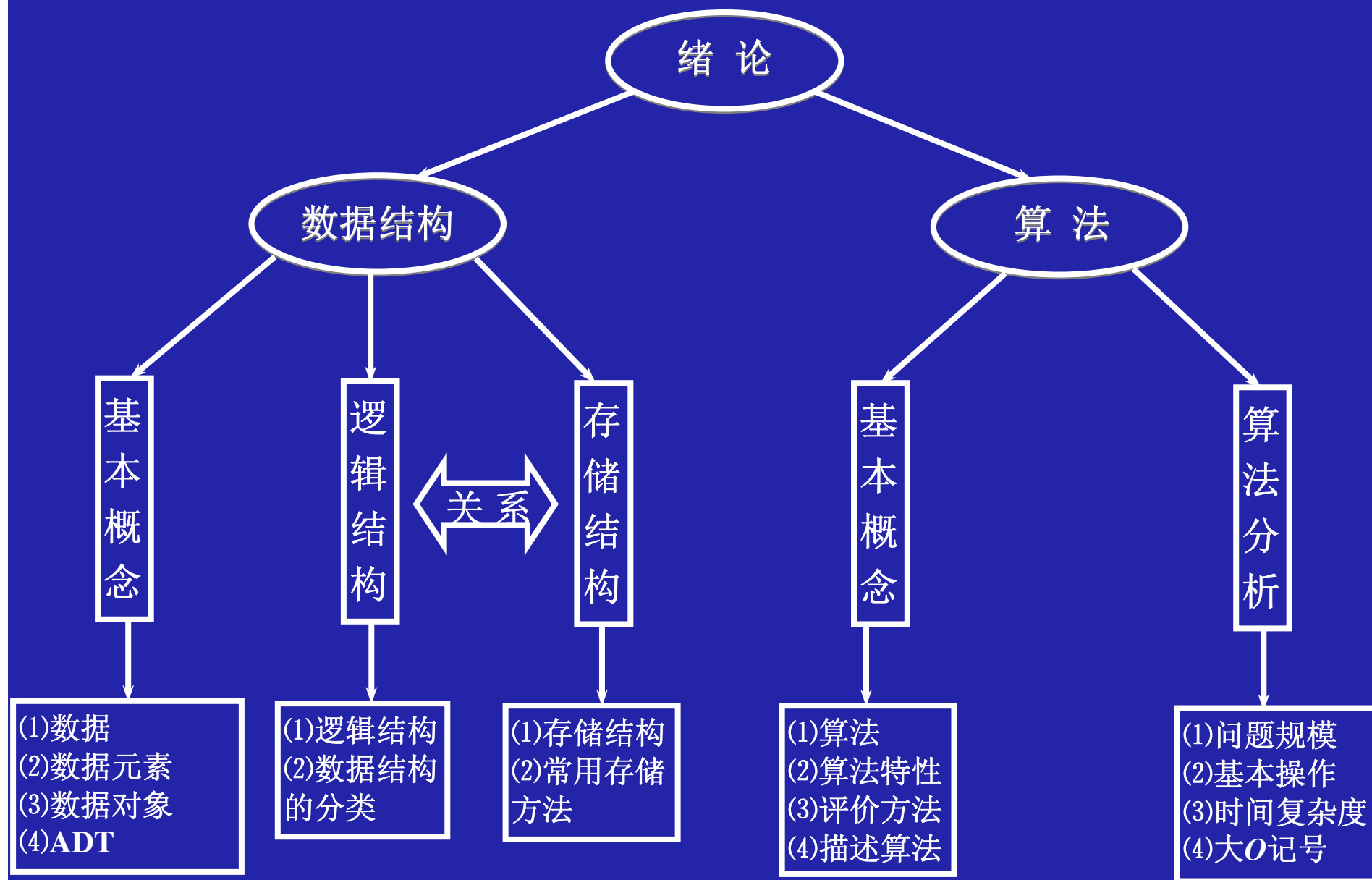
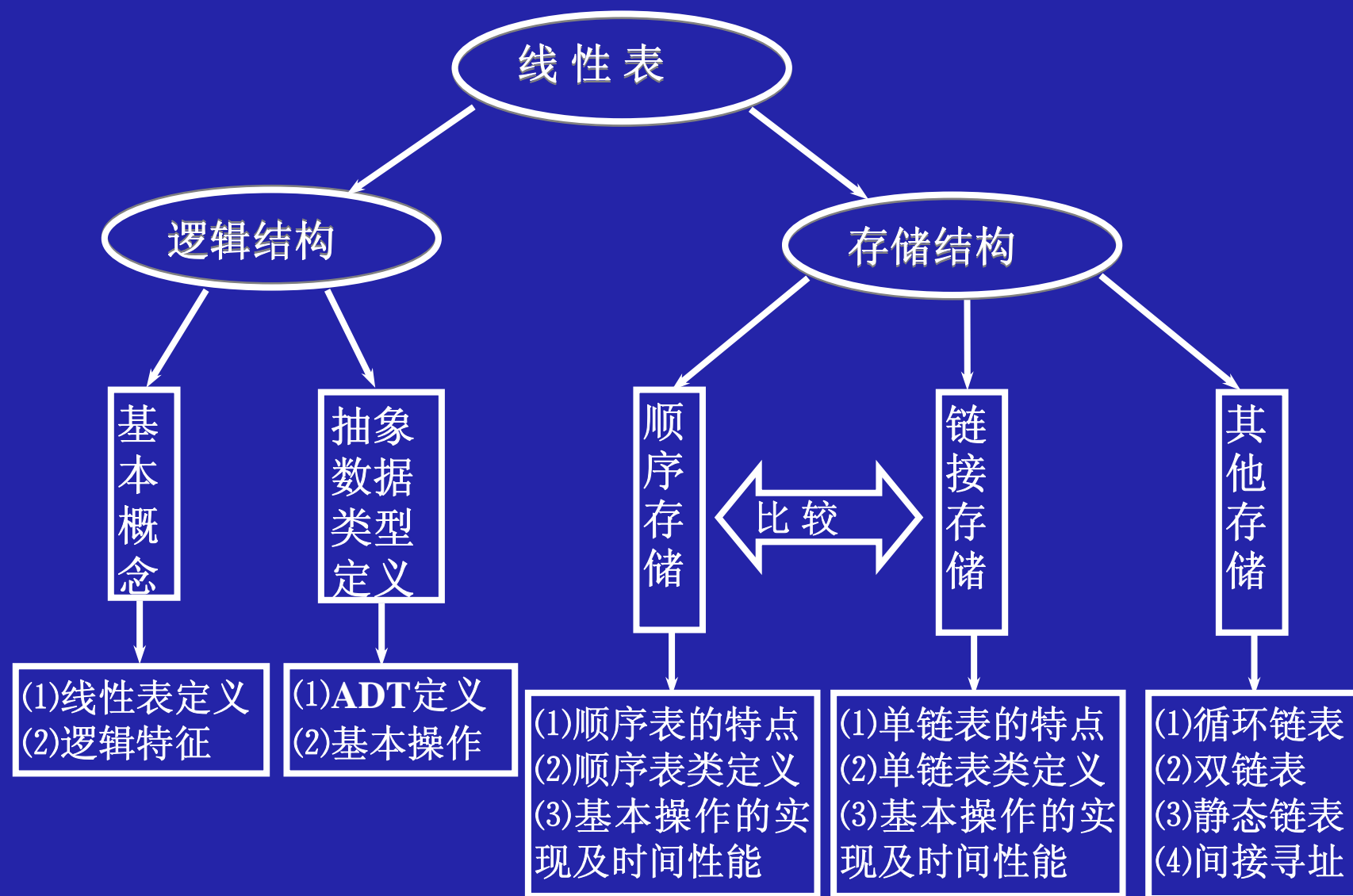


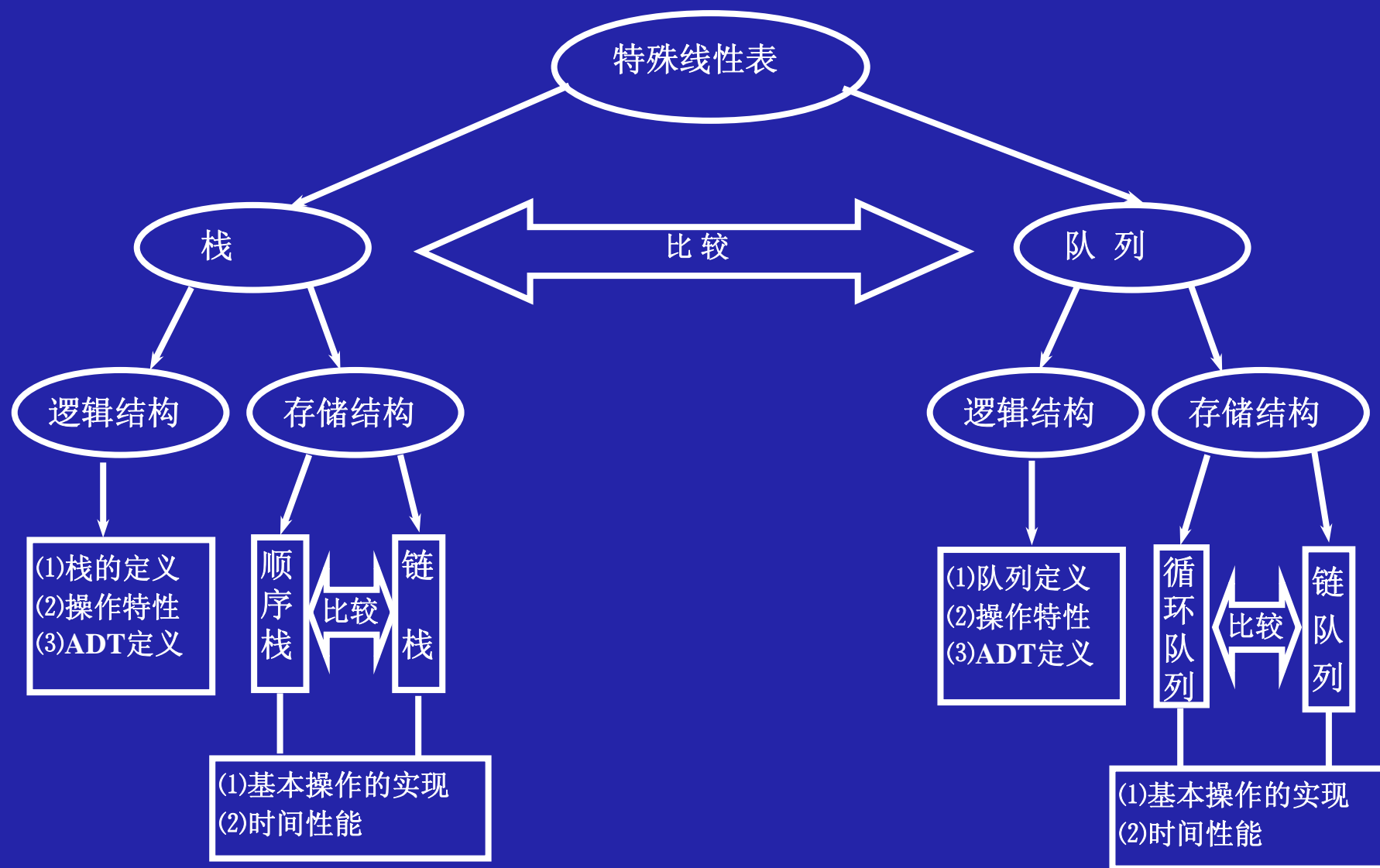
课程的知识体系

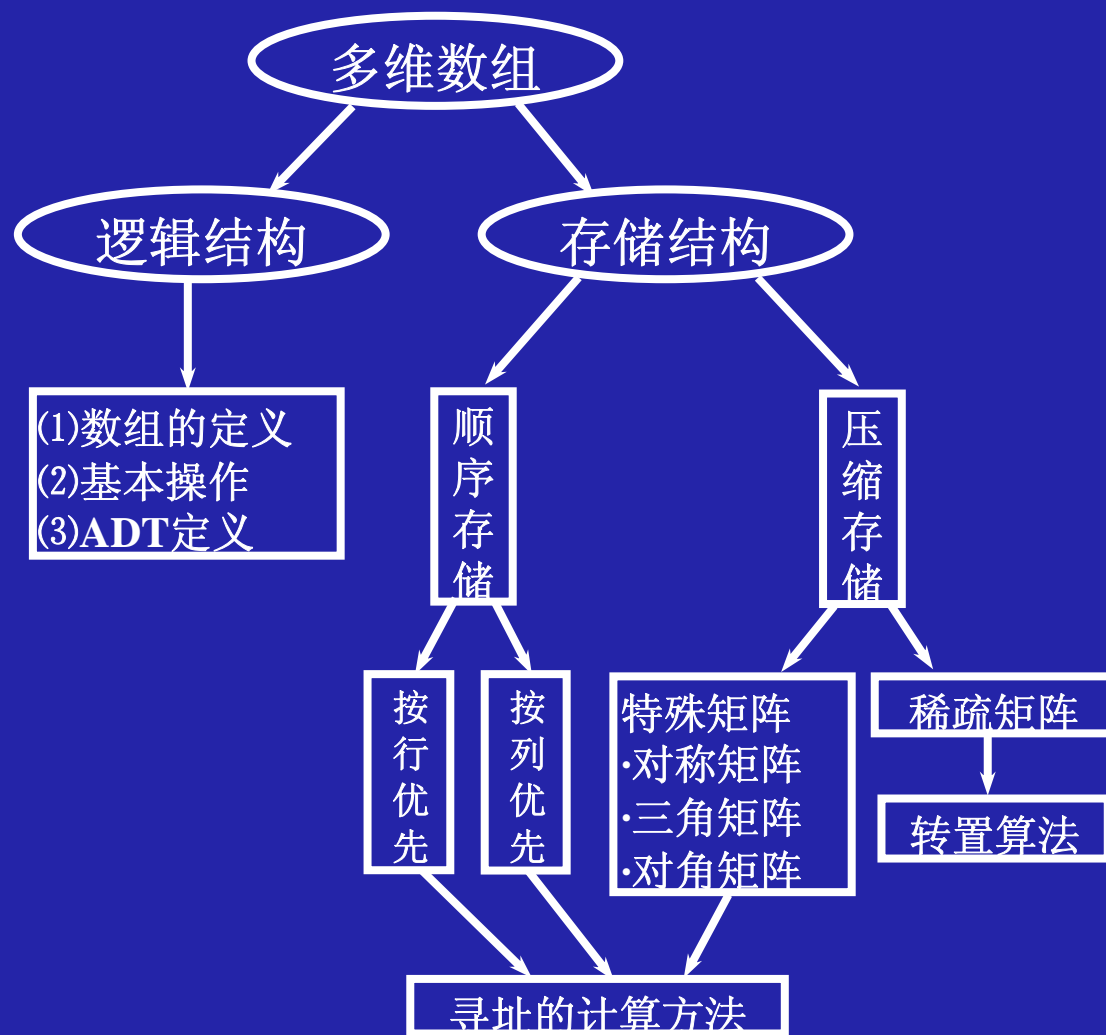


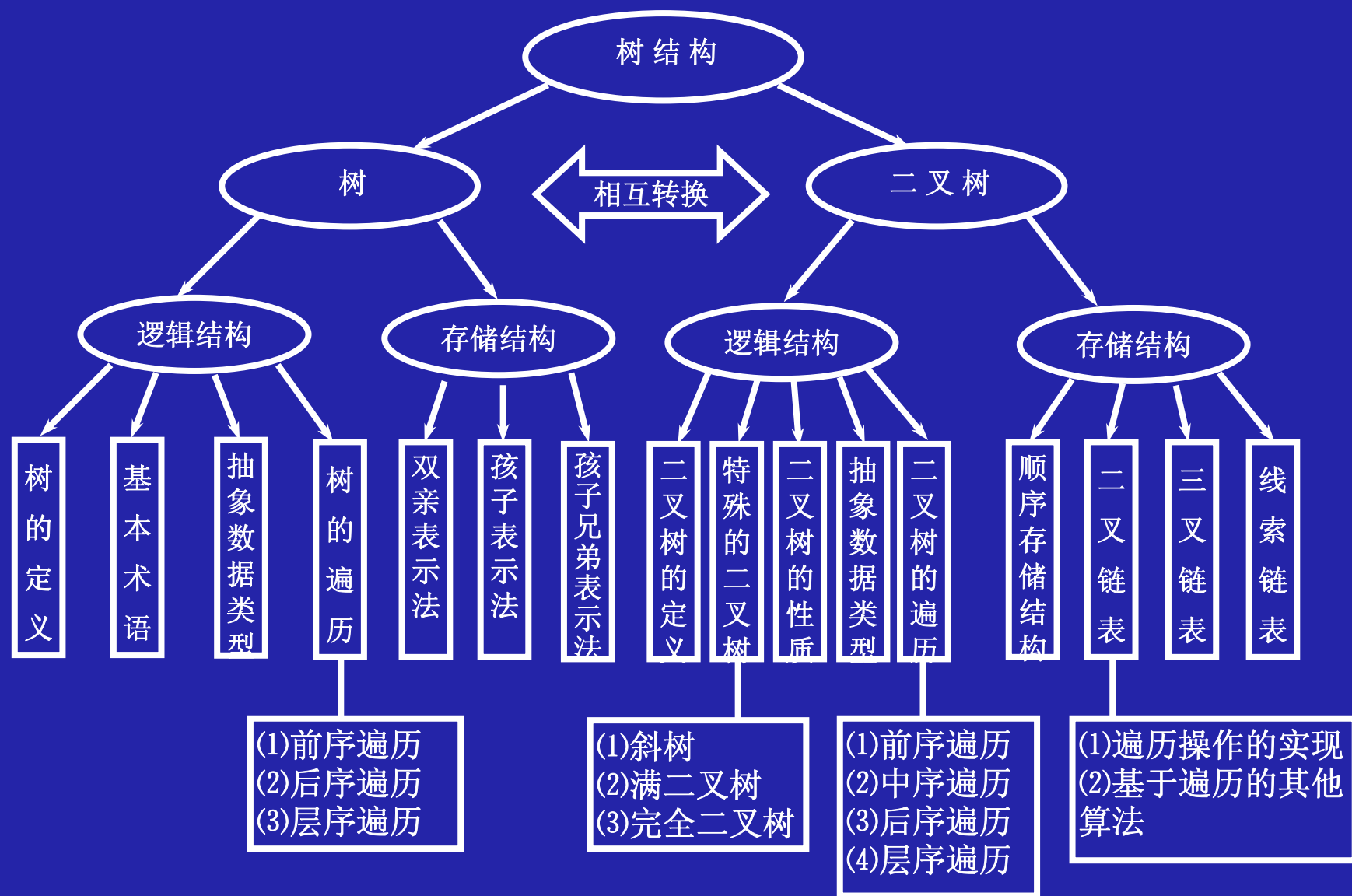
课程的核心算法：遍历（搜索、查找、检索）算法

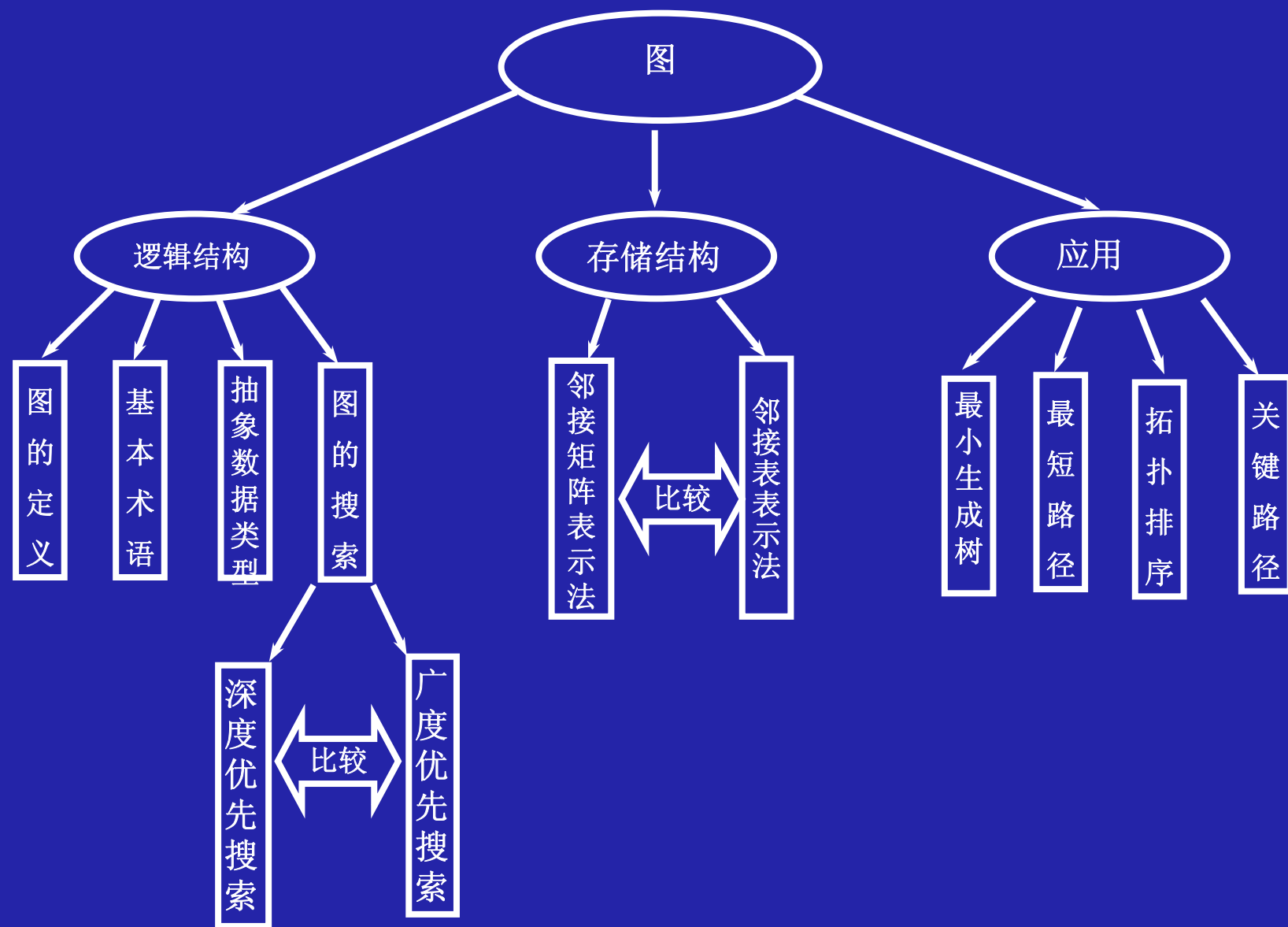












查找

- 熟练掌握顺序表和有序表的查找方法。
- 熟练掌握BST的构造和查找方法。
- 熟练掌握AVL树和B-树的结构、性质以及查找方法。
- 熟练掌握哈希表的构造方法，深刻理解哈希表与其它结构的表的实质性的差别。
- 掌握描述查找过程的判定树的构造方法，以及按定义计算各种查找方法在等概率情况下查找成功时的平均查找长度。

时间复杂度比较

排序方法	平均情况	最好情况	最坏情况
直接插入排序	$O(n^2)$	$O(n)$	$O(n^2)$
希 尔 排 序	$O(n\log_2 n)$	$O(n^{1.3})$	$O(n^2)$
起 泡 排 序	$O(n^2)$	$O(n)$	$O(n^2)$
快 速 排 序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$
简单选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$
堆 排 序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$
归 并 排 序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$

空间复杂度比较

排序方法	辅助空间
直接插入排序	$O(1)$
希尔排序	$O(1)$
起泡排序	$O(1)$
快速排序	$O(\log_2 n) \sim O(n)$
简单选择排序	$O(1)$
堆排序	$O(1)$
归并排序	$O(n)$

内容及要求

- 以线性表、树、二叉树、图等常用的数据结构的逻辑结构、存储结构和基本操作（算法）以及各种常用数据结构上实现的查找和排序算法为主要内容，
- 培养和训练对算法的复杂性的分析能力以及选择合适的数据结构对简单的应用问题进行有效算法的设计能力。

绪论

要求:

- 要求掌握数据结构中常用的各种基本概念和术语,
- 掌握算法描述和分析方法。
- 重点掌握数据结构的逻辑结构、存储结构及操作（算法）三方面的概念及相互关系,
- 难点是算法复杂性的分析方法。

内容和知识点:

1. 数据结构的基本概念和术语 (识记)

1. 1 数据、数据元素、数据项、数据结构等基本概念。

1. 2 数据结构的逻辑结构、存储结构及操作 (算法) 的含义及其相互关系。

1. 3 数据结构的两大类逻辑结构和四种常用的存储表示方法。

2. 数据结构在软件系统中的作用 (识记)。

2. 1 数据结构在各种软件系统中所起的作用。

2. 2 选择合适的数据结构是解决应用问题的关键步骤。

3. 算法的描述和分析 (领会)

3. 1 算法、算法的时间复杂性和空间复杂性、最坏的和平均的时间复杂性等概念。

3. 2 算法的时间复杂性不仅仅依赖于问题的规模, 也取决于输入实例的初始状态。

3. 3 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性。

线性表

要求:

- 线性表的逻辑结构和各种存储表示方法，以及定义在逻辑结构上的各种基本操作（算法）及其在存储结构上如何实现这些基本操作（算法）。
- 在熟悉这些内容的基础上，能够针对具体应用问题的要求和性质，选择合适的存储结构设计出相应的有效算法，解决与线性表相关的实际问题。
- 重点是熟练掌握顺序表和单链表上实现的各种基本操作（算法）及相关的时间性能分析，
- 难点是能够使用所学到的基本知识设计有效算法解决与线性表相关的应用问题

内容和知识点:

1. 线性表的逻辑结构（识记）

1. 1 线性表的逻辑结构特征。

1. 2 线性表上定义的基本操作（算法），并能利用基本操作（算法）构造出较复杂的算法。

2. 线性表的顺序存储结构（综合应用）

2. 1 顺序表的含义及特点，即顺序表如何反映线性表中元素之间的逻辑关系。

2. 2 顺序表上的插入、删除操作及其平均时间性能分析。

2. 3 利用顺序表设计算法解决简单的应用问题。

内容和知识点:

3. 线性表的链式存储结构（综合应用）

3. 1 链表如何表示线性表中元素之间的逻辑关系。

3. 2 链表中头指针和头结点的使用。

3. 3 单链表、双链表、循环链表链接方式上的区别。

3. 4 单链表上实现的建表、查找、插入和删除等基本操作（算法），并分析其时间复杂性。

3. 5 单循环链表以及单循环链表上的算法与单链表上相应算法的异同点。

3. 6 双链表的定义及其相关的算法。

3. 7 利用链表设计算法解决简单的应用问题。

4. 顺序表和链表的比较（领会）

4. 1 顺序表和链表的主要优缺点。

4. 2 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。

栈和队列

要求:

- 栈和队列的逻辑结构定义及在两种存储结构上如何实现栈和队列的基本操作（算法）。
- 要求在掌握栈和队列的特点的基础上，懂得在什么样的情况下能够使用栈或队列。
- 重点是掌握栈和队列在两种存储结构上实现的基本操作（算法），
- 难点是循环队列中对边界条件的处理。

内容和知识点:

1. 栈的逻辑结构、存储结构及其相关算法（综合应用）

1. 1 栈的逻辑结构特点，栈与线性表的异同。

1. 2 顺序栈和链栈上实现的进栈、退栈等基本算法。

1. 3 栈的“上溢”和“下溢”的概念及其判别条件。

1. 4 利用栈设计算法解决简单的应用问题。

2. 队列的逻辑结构、存储结构及其相关算法（综合应用）

2. 1 队列的逻辑结构特点，队列与线性表的异同。

2. 2 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。

2. 3 队列的“上溢”和“下溢”的概念及其判别条件。

2. 4 使用数组实现的循环队列取代普通的顺序队列的原因。

2. 5 循环队列中对边界条件的处理方法。

2. 6 利用队列设计算法解决简单的应用问题。

3. 栈和队列的应用（领会）

栈和队列的特点，什么样的情况下能够使用栈或队列。

多维数组

要求:

- 多级数组的逻辑结构特征及其存储方式，特殊矩阵和稀疏短阵的压缩存储方法
- 重点是熟悉多维数组的存储方式、矩阵的压缩存储方式、
- 难点是稀疏短阵的压缩存储表示下实现的算法。

内容和知识点:

1. 多维数组（领会）
 1. 1 多维数组的逻辑结构特征。
 1. 2 多维数组的顺序存储结构及地址计算方式。
 1. 3 数组是一种随机存取结构的原因。
2. 短阵的压缩存储（领会）
 2. 1 特殊矩阵和稀疏矩阵的概念。
 2. 2 特殊矩阵和压缩存储时的下标变换方法。
 2. 3 稀疏矩阵的三元组表表示方法及相关算法。

树与二叉树

要求:

- 二叉树的定义、性质、存储结构、遍历、线索化,
- 树的定义、存储结构、遍历、
- 树和森林与二叉树的转换,
- 哈夫曼树及其应用（哈夫曼编码与译码）、等价分类。
- 重点掌握二叉树的遍历算法及其有关应用,
- 难点是使用所学到的有关知识设计出有效算法, 解决与树或二叉树相关的应用问题。

内容和知识点:

1. 树的概念（领会）

1. 1 树的逻辑结构特征。

1. 2 树的不同表示方法。

1. 3 树的常用术语及含义。

2. 二叉树（简单应用）

2. 1 二叉树的递归定义及树与二叉树的差别。

2. 2 二叉树的性质，了解相应的证明方法。

2. 3 二叉树的存储方法、特点及适用范围。

3. 二叉树的遍历（综合应用）

3. 1 二叉树的三种遍历算法，理解其执行过程。

3. 2 确定三种遍历所得到的相应的结点访问序列。

3. 3 以遍历算法为基础，设计有关算法解决简单的应用问题。

内容和知识点:

4. 线索二叉树 (领会)

4. 1 二叉树线索化的目的及实质。

4. 2 在中序线索树中查找给定结点的中序前趋和中序后继的方法

4. 3 查找给定结点的前序前趋和后序后继并非有效的原因。

5. 树和森林 (领会)

5. 1 树和森林与二叉树之间的转换方法。

5. 2 树的各种存储结构及其特点。

5. 3 树的遍历方法。

6. 哈夫曼树及其应用 (简单应用)

6. 1 最优二叉树和最优前缀码的概念及特点。

6. 2 哈夫曼算法的思想。

6. 3 根据给定的叶结点及其权值构造出相应的最优二叉树。

6. 4 根据最优二叉树构造对应的哈夫曼编码。

图

要求:

- 图的基本概念、两种常用的存储结构、两种遍历算法以及图的应用算法,
- 重点掌握图的两种存储结构上实现的遍历算法。
- 难点是图的应用算法: 求最小生成树, 求单源最短路径, 拓扑排序, 关键路径。要求掌握这些算法的基本思想及时间性能。

内容和知识点:

1. 图的概念（领会）
 1. 1 图的逻辑结构特征。
 1. 2 图的常用术语及含义。
2. 图的存储结构（简单应用）
 2. 1 邻接矩阵和邻接表这两种存储结构的特点及适用范围。
 2. 2 根据应用问题的特点和要求选择合适的存储结构。
3. 图的遍历（简单应用）
 3. 1 连通图及非连通图的深度优先搜索和广度优先搜索两种遍历算法，其执行过程以及时间复杂性分析。
 3. 2 确定两种遍历所得到的顶点访问序列。
 3. 3 图的两种遍历与树的遍历之间的关系。
 3. 4 两种遍历所使用的辅助数据结构(栈或队列)在遍历过程中所起的作用。
 3. 5 利用图的两种遍历设计算法解决简单的应用问题。

内容和知识点:

4. 生成树和最小生成树（领会）

4. 1 生成树和最小生成树的概念。

4. 2 对遍历给定的图，画出深度优先和广度优先生成树或生成森林。

4. 3 **Prim**和**Kruskal**算法的基本思想、时间性能及这两种算法各自的特点。

4. 4 要求对给定的连通图，根据**Prim**和**Kruskal**算法构造出最小生成树。

5. 最短路径（领会）

5. 1 最短路径的含义；

5. 2 求单源最短路径的**Dijkstra**算法的基本思想和时间性能。

5. 3 对于给定的有向图，根据**Dijkstra**算法画出求单源最短路径的过程示意图。

6. 拓扑排序（领会）

6. 1 拓扑排序的基本思想和步骤。

6. 2 拓扑排序不成功的原因。

6. 3 对给定的有向图，若拓扑序列存在，则要求写出一个或多个拓扑序列。

内容和知识点:

7. 关键路径（领会）

7. 1 AOE网的概念、性质。

7. 2 路径长度、关键路径、关键活动的概念

7. 3 关键路径和关键活动性质分析

7. 4 利用拓扑排序算法求关键路径和关键活动算法的基本思想和步骤

7. 5 对于给定的AOE网，求出关键路径和关键活动

查找

要求:

- 在线性表、树和散列表上的查找方法、算法实现以及各种查找方法的时间性能(平均查找长度)分析。
- 重点掌握顺序查找、二分查找，二叉查找树上查找以及散列表上查找的基本思想和算法实现。
- 难点是二叉查找树的删除算法和AVL树上的旋转变换。

内容和知识点:

1. 基本概念（识记）
 1. 1 查找在数据处理中的重要性。
 1. 2 查找算法效率的评判标准。
2. 线性表的查找（简单应用）
 2. 2 顺序查找、二分查找的基本思想、算法实现和查找效率分析。
 2. 2 顺序查找中哨兵的作用。
 2. 3 二分查找对存储结构及关键字的要求。
 2. 4 通过比较线性表上两种查找方法的优缺点，能根据实际问题的要求和特点，选择出合适的查找方法。

内容和知识点:

3. 树的查找（简单应用）

3. 1 二叉查找树定义和特点以及用途。

3. 2 二叉查找树的插入、删除、建树和查找算法及时间性能。

3. 3 建立一棵二叉查找树的过程实质上是对输入实例的排序过程，输入实例对所建立的二叉查找树形态的影响。

4. 散列表（简单应用）

4. 1 散列表、散列函数、散列地址和装填因子等有关概念。

4. 2 散列函数的选取原则及产生冲突的原因。

4. 3 几种常用的散列函数构造方法。

4. 4 两类解决冲突的方法及其优缺点。

4. 5 产生“堆积”现象的原因。

4. 6 采用线性探测法和拉链法解决冲突时，散列表的建表方法、查找过程以及算法实现和时间分析。

4. 7 散列表和其它表的本质区别。

内部排序

要求:

- 内部排序的基本思想、排序过程、算法实现、时间和空间性能的分析以及各种排序方法的比较和选择。
- 重点掌握快速排序、堆排序、归并排序和基数排序的基本思想及排序过程,
- 难点是快速排序、堆排序、归并排序和基数排序四个排序算法的实现。

内容和知识点:

1. 基本概念（识记）

排序在数据处理中的重要性。

1. 2 排序方法的“稳定”性含义。

1. 3 排序方法的分类及算法好坏的评判标准。

2. 插入排序（综合应用）

2. 1 直接/折半插入排序和希尔排序的基本思想和算法实现，以及在最好、最坏和平均情况下的时间性能分析。

2. 2 直接插入排序中哨兵的作用。

2. 3 折半插入排序对插入排序时间复杂度的影响

2. 4 希尔排序分割待排序记录的目的、增量的选取，在子序列内如何进行直接插入排序

2. 5 针对给定的输入实例，要能写出直接/折半插入排序的排序过程。

内容和知识点:

3. 交换排序（综合应用）

3. 1 冒泡排序的基本思想。

3. 2 快速排序的基本思想和算法实现，以及在最坏和平均情况下的、时间性能分析，了解算法的稳定性。

3. 3 基准元素(划分元)对划分是否平衡的影响。

3. 4 针对给定的输入实例，能写出快速排序的排序过程。

4. 选择排序（简单应用）

4. 1 堆、小顶堆、大顶堆、堆顶等有关概念和定义。

4. 2 堆性质及堆与完全二叉树的关系。

4. 3 直接选择排序和堆排序的基本思想和算法实现，以及时间性能分析

4. 4 针对给定的输入实例，写出堆排序的排序过程。

内容和知识点:

5. 归并排序（领会）

5. 1 归并排序的基本思想和算法实现，以及时间性能分析。

5. 2 针对给定的输入实例，能写出归并排序的排序过程。

6. 基数排序（领会）

6. 1 基数排序的基本思想和算法实现，以及时间性能分析。

6. 2 针对给定的输入实例，能写出基数排序的排序过程。

6. 3 基数排序与其它几类排序的区别。

7. 各种排序方法的比较和选择（简单应用）

7. 1 通过对被排序的记录数目、记录信息量的大小、关键字的结构及初始状态、稳定性要求、辅助空间的大小、各种时间性能等方面的比较掌握各种排序的优缺点。

7. 2 根据实际问题的特点和要求选择合适的排序方法。

- 基本概念

- 文件、属性和域、关键字、主关键字、次关键字

- 文件的逻辑结构

- 文件的物理结构

- 文件的操作

- 检索方式和更新方式

- 文件的组织方式包括那些？各有什么特点？异同？适合于什么场合？

- 外部排序的概念以及与内部排序的差异
- 外部排序实现，主要依靠数据的内、外存交换和“内部归并”。
- 外部排序基本上包括相对独立的两个阶段：初始归并段的形成和多路归并。
- 磁盘文件的外部排序主要研究的技术问题是：
 - (1) 如何进行多路归并以减少文件的归并遍数；
 - (2) 如何巧妙地运用内存的缓冲区使I/O和CPU尽可能并行工作；
 - (3) 根据外存的特点选择较好的产生初始归并段的方法。
- 磁带文件的外部排序的主要方法是：
K路平衡归并排序和多阶段归并排序