

期末考试题型

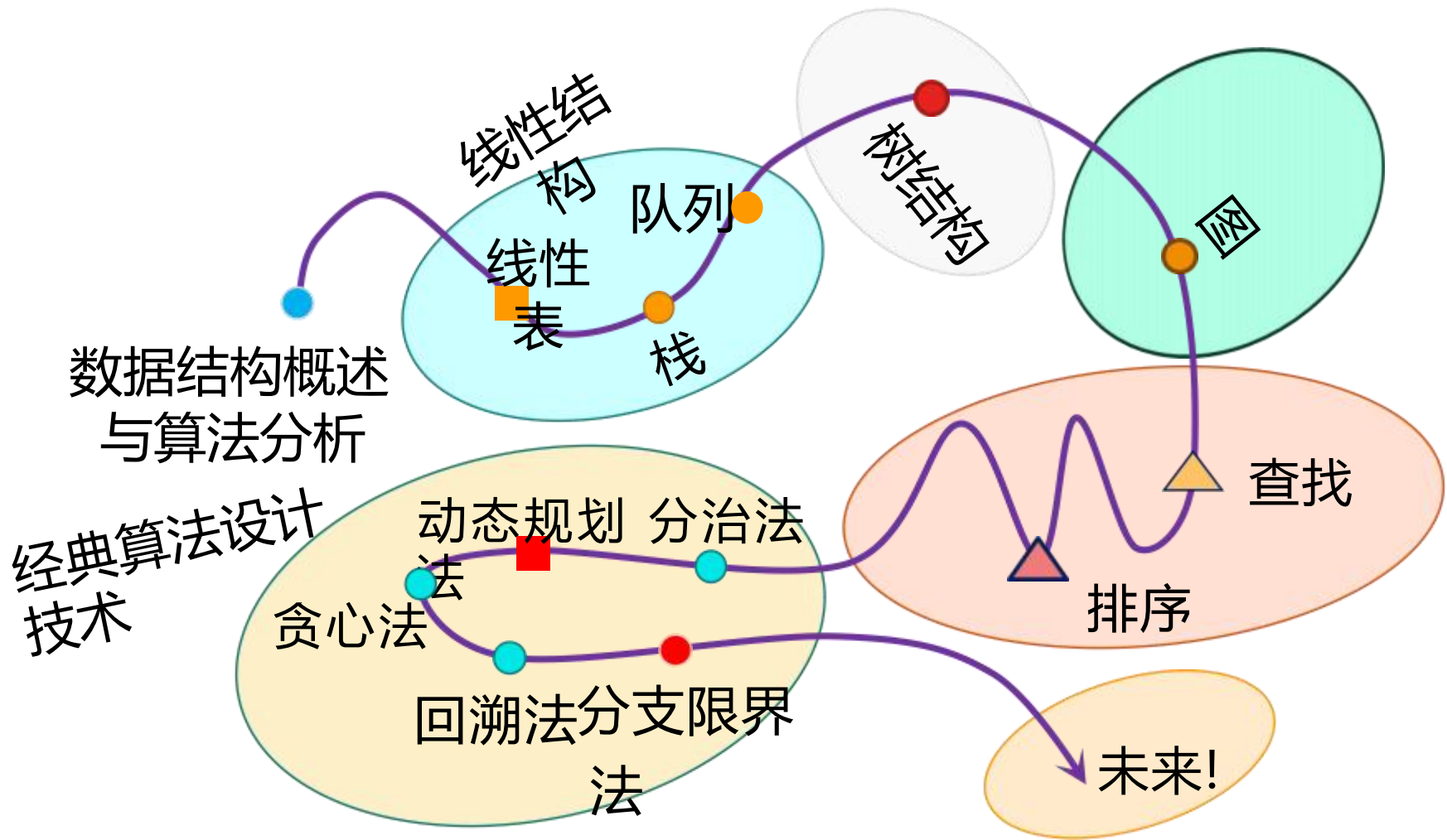
- 应用题
- 算法设计题

闭卷考试 **120**分钟

内容

- ✓ 基本知识
- 基本技能
- 题型及样题

知识点思维导图



考核目标

- 掌握线性结构、树形结构和图结构等常用数据结构的抽象数据类型、物理实现、主要的算法，形成一个程序员的基本数据结构工具箱。
- 掌握排序和查找等算法，了解五大经典算法设计技术的基本思想，掌握算法效率的度量分析方法。
- 能根据实际需求的需求，选择合适的数据结构来表示、存储和处理。针对复杂工程问题所要求的资源限制和算法，权衡时间与空间开销，设计更有效的解决方案。
- 训练计算思维能力、算法设计和分析能力和程序设计与实现能力
- 从理论、抽象、设计 的角度来解决问题

数据结构概述和算法分析

- 抽象数据类型
- 数据结构
- 算法的代价
 - 时间代价
 - 空间代价
- 渐近算法分析
- 算法的最佳情况、最差情况和平均情况
- 上限、下限
- 大O、大 \mathbf{Q} 和大 Θ 表示法
- 定义估算法
- 化简法则
- 递归算法的分析

概念 **Concept**

设计 **Design**

实现 **Implement**

应用 **Application**

数据结构部分-线性表、树和图

- 线性结构
 - 线性表
 - 线性表**ADT**
 - 线性表的物理实现
 - 线性表的应用
 - 栈 (**LIFO**)
 - 栈的**ADT**
 - 栈的物理实现
 - 栈的应用
 - 队列 (**FIFO**)
 - 队列的**ADT**
 - 队列的物理实现
 - 队列的应用
 - 树型结构
 - 树结构的性质
 - 树和二叉树的**ADT**
 - 树的物理实现
 - 二叉树的遍历算法
 - 二叉树遍历算法的应用
 - **BST**树的特性
 - **BST**的实现
 - 堆树的特性
 - 堆树的实现
 - 哈夫曼编码树的特性
 - 哈夫曼编码树的实现
 - 网状结构
 - 图
 - 图**ADT**
 - 图的物理实现
 - 图的遍历算法
 - 拓扑排序
 - 关键路径
 - 最短路径算法
 - 最小生成树算法
- 概念**Concept**
设计**Design**
实现**Implement**
应用**Application**

查找和排序部分

- 查找

- 基于线性结构的查找

- 顺序查找
 - 二分查找
 - 自组织线性表

- 基于树型结构的查找

- **BST**的查找
 - **AVL**

- 散列

- 散列函数
 - 冲突解决策略

- 排序

- 基于相邻元素比较的排序

- 冒泡排序
 - 选择排序
 - 插入排序

- **Shell**排序

- 快速排序
 - 归并排序

- 堆排序

- 基于计算的排序

- 基数排序
 - 排序问题的下限

排序算法思想、
排序过程、
排序算法的佳
情况、平均情况
和最差情况的复
杂度、
排序算法的稳定
性

概念**Concept**

设计**Design**

实现**Implement**

应用**Application**

算法设计技术部分

算法

分治与递归

二分搜索、归并排序、快速排序、基于BST的查找
二叉树的遍历、最大整数乘法，最近点对问题等

动态规划

Floyd算法、0-1背包问题，最长公共子序列

贪心算法

Prim算法、Kruskal算法、Dijkstra算法、哈夫曼编码、背包问题、货物装载问题等

回溯法

哈密尔顿回路问题、0-1背包、货物装载、N皇后问题、子集和问题等

分支限界法

单源最短路径、0-1背包、货物装载问题等

算法设计技术

- 分治法与递归
 - 基本思想
 - 基本步骤
 - 能解决问题的特征
 - 分治法的实现
 - 分治法的应用
 - 分治法复杂度的计算
 - 递归公式
 - 主定理法
- 动态规划法
 - 基本思想
 - 基本步骤
 - 能解决问题的特征
 - 最优子结构性质
 - 递归地定义最优值
 - 复杂度的计算
 - 动态规划算法的应用
 - 动态规划算法的实现
- 贪心法
 - 基本思想
 - 基本要素
 - 贪心选择性质
 - 最优子结构性质
 - 与动态规划法的异同
 - 贪心法的应用
 - 贪心法的实现
 - 贪心法正确性的证明
 - 复杂度的计算

概念 **Concept**

设计 **Design**

实现 **Implement**

应用 **Application**

算法设计技术

- 回溯法
 - 基本概念和基本思想
 - 解空间树
 - 子集树
 - 排列树
 - 基本步骤
 - 限界函数与剪枝
 - 回溯法的算法框架
 - 回溯法的应用
 - 回溯法复杂度的计算
- 分支限界法
 - 基本思想和基本概念
 - 与回溯法的联系与区别
 - 搜索策略
 - 限界函数与剪枝
 - 分支限界法的算法框架
 - 队列式(**FIFO**)分支限界法
 - 优先队列式分支限界法
 - 分支限界法的应用
 - 复杂度的计算

概念 **Concept**

设计 **Design**


实现 **Implement**


应用 **Application**

内容

- 基本知识
- ✓ 基本技能（能力点）
- 题型及样题

算法分析—能力点小结

- 
- 理解算法复杂度符号的含义，能够专业的阐述其内涵
 - 算法复杂度、增长率、 $T(n)$
 - O 、 Ω 、 Θ
-

- 
- 掌握基本的算法分析方法，能够分析简单算法的性能
 1. 求 $T(n)$, 使用化简规则求 (时间) 复杂度
 2. 使用定义法分析 (时间) 复杂度
 3. 使用迭代递推法进行递归算法分析 (2.8.1 递归算法分析)

黑色是基本能力
红色是提高能力

数据结构概述—能力点小结

- 数据结构的含义
- 抽象数据类型的含义



黑色是基本能力
红色是提高能力

线性结构—能力点小结




- 线性表、 栈和队列特点
 - 线性表、 栈和队列的两种物理实现方式（顺序表和链表）
的特点
-




- 线性表、 栈和队列ADT的设计和表示
- 基于线性表、 栈和队列ADT设计简单问题的算法
- 基于线性表、 栈和队列ADT设计较复杂问题的算法
- 基于顺序表实现线性表、 栈和队列
- 基于链表实现线性表、 栈和队列

黑色是基本能力
红色是提高能力


树型结构—能力点小结


- 
- 树的特点和性质
 - 树的物理存储方式
 - 二叉树的遍历
-

- 
- 树的动态左子结点/右兄弟结点表示法转换
 - 根据给定的二叉树写出层次、前序、中序和后序遍历序列
 - 根据二叉树的两种遍历序列，构造出对应的二叉树
 - 二叉树的遍历算法实现
 - 基于二叉树递归或遍历的思想设计简单问题的算法
 - 基于二叉树递归或遍历的思想设计较复杂问题的算法
 - 二叉树的基本操作的实现
 - 二叉树的构建算法

黑色是基本能力
红色是提高能力

特殊树结构—能力点小结

- 
- BST树特征
 - 堆树特征
 - Huffman编码树特征
 - 并查集和根树特征
-

- 
- BST的构建方法
 - BST的递归和迭代实现方法
 - 堆的构建方法
 - 堆的实现方法
 - Huffman编码树的构建方法以及编码
 - 利用UNION/FIND算法解决等价类问题

黑色是基本能力
红色是提高能力

图结构—能力点小结

- 图的特点和性质
- 图的物理存储方式邻接矩阵或邻接表特点
- 图的遍历算法思想和性能

- 根据图构造邻接矩阵或邻接表（邻接矩阵或邻接表构造图）
- 基于邻接矩阵或邻接表实现图的基本操作
- 图的深度优先搜索序列、深度优先搜索树
- 图的广度优先搜索序列、广度优先搜索树
- 基于图的ADT设计算法解决图的应用问题
- 根据给定的图构造顶点的拓扑序列、拓扑排序算法
- 基于数据结构设计图遍历伪代码
- 应用图的遍历思路设计图的简单问题算法
- 根据给定的图求关键路径

黑色是基本能力
红色是提高能力

经典图问题—能力点小结

- 最短路径问题的内涵
- 经典图的最短路径问题算法思想和特点
- 最小支撑树的内涵
- 经典图的最小支撑树问题算法思想和特点

- 用Dijkstra算法求单源最短路径
- 用Floyd算法求每对顶点间的最短路径
- 按照Prim算法构造图的最小支撑树
- 按照Kruskal算法构造图的最小支撑树
- Dijkstra算法伪代码和Floyd算法伪代码
- Prim算法伪代码和Kruskal算法伪代码
- 应用图的最短路径问题求解思路设计图的相关问题算法
- 应用图的最小支撑树问题求解思路设计图的相关问题算法

黑色是基本能力
红色是提高能力

查找—能力点小结

- 查找概述
- 各类经典查找方法的思路和特点、性能

-
- 顺序查找的实现
 - 二分查找的实现
 - 结合分治法和递归，掌握二分法的算法思想、算法伪代码和复杂度分析
 - BST树的查找过程和性能计算 (ASL)
 - AVL树的构建、插入和删除
 - 散列表的构建
 - 在散列表中插入、检索和删除记录 and 性能计算

黑色是基本能力
红色是提高能力

排序—能力点小结

- 排序概述
- 各类排序方法的思路和特点、性能
 - 插入排序、冒泡排序和选择排序
 - **Shell排序、快速排序、归并排序、堆排序**
 - **分配排序和基数排序**

-
- 给出一组关键码，写出按某种排序算法进行排序的每一趟的结果
 - 简单和经典排序方法的伪代码
 - **应用排序的思想设计（优化）相关问题算法**

黑色是基本能力
红色是提高能力

内容

- 基本知识
- 基本技能
- ✓ 题型及样题
 - ✓ 参见经典例题网站上各章的课程考试真题