课程总结

- 课程的目标和意义
- 问题求解，算法和数据结构
- 主要数据结构

  结构、性质和实现
- 重要算法

  解决的问题/基本思想/实现细节
- 复习和考试

课程的目标和意义

- 用计算的方式解决问题时，最基本的问题有两个
  - 如何安排（组织）和管理好计算中需要处理的信息
  - 如何设计出能够完成工作的计算过程

  前者就是数据结构的问题，后者是算法问题

  N. Wirth 给出了著名的公式：算法 + 数据结构 ＝ 程序设计
- 在理解了基本程序设计技术和所用工具（编程语言，这里是 **Python**）之后，下一步的问题就是如何组织数据和设计算法

  这些就是本课程的基本内容
- 作用：
  - 掌握一些数据组织的基本技术，理解它们的性质
  - 通过实例和讨论，了解一些基本算法和算法设计的基础知识
  - 通过练习和大作业，取得在实际中应用计算解决问题的初步经验

主要数据结构

- 本课程讨论的基本问题：
  - 算法，数据结构，程序，Python 程序设计
  - 时间复杂性，最坏情况，平均情况，分期付款式的复杂性
  - 空间复杂性，最坏情况，平均情况
- 需要理解算法复杂性的意义
  - 经常关心最坏情况时间复杂性，有时也关注平均时间复杂性
  - 掌握简单的算法复杂性分析技术（加法规则和乘法规则）。空间复杂性一般更简单，通常可以从算法描述直接得到
- 讨论了 Python 的一些内部细节和高级编程技术，如
  - Python 类和面向对象程序设计
  - Python 内置数据结构的性质：list 和 dict
  - Python 的一些编程技术，等等

主要数据结构

- 典型的基本数据结构
  - 线性表：简单线性结构
  - 栈，队列，优先队列：用于保存中间信息的存储结构
  - 树和二叉树：分层结构
  - 图：元素间存在复杂关系的结构
  - 散列结构（在顺序表基础上加一个散列函数）
- 一些结构有清晰的操作集合，如栈、队列、优先队列。其他结构有一批基本操作，可根据需要扩充。常见操作：创建，检查（空/满判断等），访问/设置/增/删元素，访问/使用元素间关系，修改结构等
- 数据结构的基本实现方式：
  - 顺序：元素顺序排列，用排列顺序和位置表示元素间关系
  - 链接：显式记录元素间的关系，利用动态存储管理的功能实现
  - 复杂的数据结构常采用两种或多种方式的组合

数据结构

- 复杂结构实例：
  - 拉链法散列、邻接表：一个顺序表和一组链接表
  - 动态顺序表：基本表示为顺序，利用链接的灵活性
- 实现结构的设计需要考虑：
  - 各种基本操作的需要，使操作的实现方便
  - 考虑操作效率和存储效率
- 数据结构的典型应用：字典和检索。目标：
  - 最重要的考虑是提供高效的检索操作
  - 支持高效动态操作，插入/删除。静态字典不考虑这方面

重要算法

- 课程中仔细介绍了一批重要的基础算法，是本课程的重要内容。对一个算法的功能和过程的理解，可分为三个层次：
  - 该算法解决的问题，算法设计的基本思想
  - 基本过程（能按算法描述的过程处理简单实例，得到相应的解）
  - 理解和掌握算法实现细节，包括所需辅助数据结构，能编程实现
- 一些算法只要求理解基本思想和基本过程（前两层为考试要求）：
  - 字符串 KMP 匹配算法， Huffman算法，几个图算法
- 需要理解算法的性质（优劣、比较）
  - 时间复杂性和空间复杂性
  - 有些问题存在复杂性低的算法，为什么复杂性较高的算法还用？（实现复杂性/常量因子/其他性质/问题规模）
  - 解决具体问题的算法可能有一些具体性质

算法

重要算法

- 二叉树周游算法
- 宽度/深度优先搜索和周游
- **Huffman** 算法、**Huffman** 树和 **Huffman** 编码
- 二叉排序树检索、插入和删除
- 堆的筛选算法，堆的构造
- 图的基本算法
  - 最小生成树（**Prim**算法和**Kruskal**算法，解决同样问题）
  - 最短路径（**Dijkstra**算法和**Floyd**算法，解决不同问题）
  - 拓扑排序，关键路径
- 排序算法（一组算法，解决同一个问题）

算法和数据结构总结

- 排序：一个重要问题，一些解决问题的想法，一批特征各异的算法
  - 各种算法的基本思想，如：不断扩充已排序序列（插入正确位置，选择合适元素）；把大工作分解为部分（如何分，怎样由部分的结果得到整体结果）；基本过程（调整中的不变关系）
  - 时间和空间复杂性（平均和最坏）
  - 稳定性（什么是稳定性？意义何在？）
  - 不同算法的特性：时间和空间复杂性、算法的复杂性，稳定性，适应性
- 不作为考试要求的内容
  - 最佳二叉排序树，图算法的复杂性（与实现细节有关。可以把算法的复杂性定义为其最佳实现能达到的复杂性）
  - 简单介绍的内容，如多分支排序树部分（包括 **B** 树和 **B+** 树）

复习/答疑/考试

- 考试时间是 1 月 8 日上午
  - 两小时，闭卷
  - 考试中只带必要的文具，带学生证
  - 注意学校规定：迟到超过15分钟不得入场；开考30分钟后方可交卷
  - 未经监考老师许可，不得离开自己的座位（交卷除外）
- 考前答疑
  - 地点：理科一号楼 1479 教室
  - 时间： 1 月 6 日，15-17点；1 月 7 日，9-11 和 15-17 点
  - 其他时间可以到我办公室

    理科一号楼 1480

试题情况和要求

- 填空和选择。根据题目要求做
- 问答题，与基本的概念和技术有关的问题

  可能要求给出简要的回答，或做一些分析/比较/讨论
- 操作题，如
  - 根据定义和性质完成某些简单计算
  - 手工完成一些操作，对具体实例说明算法的过程和结果
  - 画图说明一些数据结构的情况
  - 操作题应该有过程，说明如何得到结果
- 编程题，数据结构描述和算法实现，写出 Python 程序。注意：
  - 需要说明程序处理的结构，相关的数据表示
  - 应给出适当解释，尽可能写清楚，使改卷时容易辨认
- 综合题，可能包含上面几类题目的要素