

前言

大学计算机基础课程是计算机学院承担教学任务，在北航学院托管开设的信息大类基础课程之一，是全体理科大类及强基计划、拔尖计划相关专业的学生在大一时应当修习的一门基础课程。课程主要内容涵盖计算机组成原理、编译原理、Python 基础语法、程序控制结构、算法基础、图表绘制、数据分析、GUI 设计等内容。课程内容丰富，基本涵盖了理科大类对口各专业在未来使用计算机作为工具，在实际科研或工作中解决问题的需求，或是提供了较好的入门指导工作，可以培养学习者的计算思维和运用计算机解决具体问题的能力。无论是否有计算机使用经验，是否有程序编写或脚本编写经历，在学习本课程后都将具有相应的能力。

具体的课程内容与任务方面。课程主要包括 2 次讨论课，10 次课内实验和 1 个大作业，所有题目都是助教同学利用寒假时间全新准备的。具体内容见下。

两次讨论课主要的内容是对图论（七桥问题等的理解，最短路径问题等）、最优化问题（枚举法、匈牙利算法等）等特定、相对较为复杂的算法与模型的理解和应用，由每个人独立思考，而后在小组内讨论并向全班汇报解决思路与方案，课后每个人需要形成**独立的实验报告**。

课内实验是跟随理论课程授课内容，对理论课上所讲授的知识进行上机运用的过程环节。课内共有 10 次实验，课内实验难度不是很大，最最重要的是**独立完成**（包括提交到平台上的代码和实验报告）。

具体实验安排如下：

0. Python 编程环境的使用
1. Python 基本语法
2. 程序控制结构与列表
3. 问题的描述—基本数据结构（字符串、字典）
4. 问题的描述—自定义数据结构（栈、队列）

5. 基本算法设计与实现（枚举，递归）
6. 较复杂算法设计与实现（贪心法，动态规划）
7. Python 实现数据图形绘制
8. 插值、拟合计算
9. Pandas 数据分析与 GUI 设计

其中，前几次实验（实验 0-实验 2）的难点在于对 Python 最基础的语法的理解与运用，难在入门。这些内容并不是很难，甚至可以称得上非常简单。只是可能较为新鲜，此前没有接触过，才会带来一定的困扰。只要实际上手写过一两个程序，这部分内容就可以很轻松地掌握。

在基础数据结构和算法部分（实验 3-实验 6），对知识本身的理解与运用则会成为相对困难的内容，但本课程的目的旨在培养运用与实用的思维，并不在于要求对这些算法有多么深刻的理解，也不会要求大家去 leetcode 等算法训练平台做大量的题来理解和巩固，当然，少量做一些题目，来提高和巩固自己也是欢迎的。事实上，贪心法和动态规划都是计算机科学发展到现在，称得上是最基本、最简单而常用的计算机算法了，所以其难度并不是那么大，还请大家多多尝试学习和理解。

最后的几次实验（实验 7-实验 9），我认为这些是应用性较强的实验。直白来说，在这部分实验中，你并不需要知道多少原理，你要做的只是拿来几个模板，照猫画虎，按部就班地搭建“积木”，完成任务就可以了，并没有多少难度。但鉴于 Python 是一门脚本语言，从笔者的实际经历来看，这些内容在未来的实际科研和应用中，反而是非常非常有用的。

大作业通常在学期中后段布置，是一组具有不同难度系数，面向不同实用场景的实际问题的，较为系统和工程化的解决方案，是对整个学期课程所学的一次总体性的应用。在完成大作业的过程中，往往需要跟随实验指导，自学一定量的知识。大作业题目具有一定的复杂性，所以也需要投入一定的时间去完成。通常在大作业后也会安排申优答辩环节，可以选择参加，需要花费一定的时间和精力

去准备。

完成对课程内容和总体任务的介绍后，让我们回到编纂手册的目的上来。正如前面所说，本课程授课过程面临时间紧，任务重的困难，对于少部分没有计算机基础或使用经验的同学来说，学习这门课程的过程中可能会面临较大的困难。这往往体现在开展实验实操时进度不佳，或面对的困难较多而疲于应对。而苦于授课教师和助教精力有限，分身乏术，在提供实验指导方面可能会有覆盖不到的地方，对部分同学的学习带来一定的压力，甚至是阻力。同时，实际线下实验指导时我们通常以口述方式进行指导，难以在讲述过程中配合相应的代码，最好的情况也是配合同学当前所写的代码进行辅导。但这对描述问题和思路上时不利的。通常，配合代码示例的形式对问题和思路进行描述往往可以使读者（或者听众）对问题和思路形成更好的理解。因此，我选择编纂本手册，一方面，是让同学们在每次实际进行实验前，通过学习实际题目和代码的方式，能够能对这一部分的内容做大致了解，做到“手上有枪”；另一方面，我作为课程的修习者，以学生的视角编纂一份手册，也是让大家能有一个相较于教材来讲，方便速查思路的材料。教材上的内容非常全面，高屋建瓴，但对于速查还是带来一定的困难，与同学们学习的思路上也不一定完全一致，而同龄人之间总是更容易找到相同的思路，产生一些共鸣。

目录

前言	i
第一章 Python 编辑及运行环境配置	1
1.1 Anaconda 的下载与安装	1

本书中所有代码，在 Windows 11 23H2 及 macOS 14.2.1 23C71 arm64 上以 Anaconda 23.11.0, Python 3.12.0 作为环境运行测试通过。具体第三方库版本见下：

1. scipy
2. numpy
3. matplotlib
4. pandas

第一章 Python 编辑及运行环境配置

1.1 Anaconda 的下载与安装

在本课程中，我们选择使用 anaconda 作为 Python 的集成开发环境。一方面是因为 anaconda 在安装时不仅仅能够安装一个 python 解释器，其也会同时安装 matplotlib, scipy, numpy 等多个常用的（也就是本课程中会用到的）几个第三方库。另一方面，anaconda 提供 Spyder 这一高集成度，便于初学者学习和调试的 IDE，可以提供给我们“开箱即用”的体验。

如果你想要挑战自己，当然可以自行到 Python 项目官网¹上自行下载和安装 Python 解释器，而后使用 pip 包管理工具安装我们所需的各类第三方库或者第三方组件（包括 Spyder，也可以通过 pip 来自动安装）。但在正式开始教程之前，需要强调，作为初学者，请不要同时安装纯净 Python 解释器和 anaconda，即不要让使用下载的 Python 安装包所安装的 python 解释器同 anaconda 在同一台电脑上并存，因为此时你的电脑里将同时有两个 python 运行环境，且这两个运行环境没有通过环境管理器进行管理，其激活次序，即运行 python 脚本文件时正在使用的 python 解释器，或是尝试使用 pip

¹<https://python.org>

安装第三方库所指向的环境不易明确。总之，作为课程的初学者，也是本手册后面的安排，我们将基于 anaconda（conda+python 解释器）进行介绍，所有代码也将会在这个环境下进行运行。

1.1.1 Anaconda 的下载