

# 前言

## 缘起

范文澜曾云“板凳要坐十年冷”，古文里也常常可见“十年寒窗”，而Peter Norvig更是写过一篇异曲同工的妙文*Teach Yourself Programming in Ten Years*.<sup>1</sup> 尽管一般人不可能用十年去培养非常专业的功底，但我们希望在有限的学习时间内培养读者的基本技能，并为他们打开程序设计这扇神奇之窗。

那么如何尽快学会搭建程序呢？乐高积木为我们提供了一种很好的思路，只需使用基本的“积木式”组件便可迅速完成程序设计。抽象数据类型正是这样的积木，而C++的标准模板库(STL)已为我们准备好了。

在学会组建程序的基础上，我们要求从算法角度分析效率，而抽象数据类型的简约性更利于在宏观上尽快给出优良的方案设计。因此，贯穿全书的主线是抽象数据类型的选择、使用和组合。

我们特别强调在抽象数据类型选用时必须以算法分析为导向，以算法效率为准绳。对于以不同抽象数据类型为要素的实现方案，其选择标准是完成整个方案所需的时空开销。此外，我们还会关注同一抽象数据类型在不同数据结构实现下的性能差异。在上述思想指导下，本书力求给出一种全新的模式，让读者尽快学会高效使用抽象数据类型，最终为后续的计算设计课程打下坚实的基础。

## 特色

首先，我们希望培养读者的算法思维乃至计算思维。在长期随意手工计算的影响下，许多人很自然地将自己的行为习惯迁移到算法设计中，其表现有多种：例如操作方式天马行空，处理数据时充满了人的跳跃性，而毫无“机械式”的章法(或者说“系统化”的处理方式)；又如侧重对小规模具体实例优化，而不考虑一般情况。实际上，如今我们处理的许多问题会涉及巨量的数据，如何高效计算则成为关键问题，要突破必须及早培养计算思维。

其次，本书力图破除很多人“只见树木，不见森林”的程序编写思路。要转为从抽象数据类型的角度思考问题，而不至于过早陷入代码细节中。全书对于数据结构的讲解侧重于如何用好抽象数据类型的各种操作，特别是每项操作的时空效率，而这正是现代算法设计对数据

---

<sup>1</sup> 事实上，本书第1版面世至今已有五年多，不少内容需要修订和完善，正好验证了上述观点。

结构知识的要求。至于传统数据结构教程强调的各种数据结构实现则不是我们关注的重点,实际上实现类库的工作相当专业,若非数十年功力绝难完成,君不见迄今为止能经得起推敲的模板库寥寥无几。从这个角度看,学好抽象数据类型并能灵活运用才应该是我们追求的目标。因此,本书着重使用STL类库解决实际问题,让读者能尽快看到数据结构的应用,而不是建立空中楼阁。

我们还将突出深入计算机内部的思想,尽量让读者把握各个时刻计算机中的实际数据情况,不妨称为 $\{0, 1\}$ 思维。这种从底层去探究算法运行本质的方法不但有助于理解算法运行情况,更能提高算法设计水平。例如初学者按照此角度可时时提醒自己要按照“机械式”方式给出步骤,而专业人士则能在纷繁复杂的数据和算法步骤中找到规律进而优化。通俗地说,要把自己想象成计算机,这样身临其境方能参透奥秘。

在许多教材中,算法只是观念上的实现,从而导致教材自身不注重具体代码的编写,即便包含代码也未精化。我们希望能给出清晰高效且具备良好代码风格的C++语言实现,并尽量兼顾学术前沿和工程实践。因此本书在第1版所创建的DSAD项目(<https://github.com/xiexiexx/DSAD>)会继续更新完善,读者可获取源代码以及全书勘误表。另外,为了便于修订,书中很多程序以在线代码形式呈现(配有网址并附二维码),阅读时以在线版本为准。

## 重构

第2版的修订融入了这几年来的一些教学思考,特别是“集合”抽象数据类型和迭代器区间的理念。为此我们重写了第2章并以此为核心重构了全书的写作。

- 泛型抽象: 从集合的观点出发,再配合迭代器,可以将很多算法抽象化,并能方便地转换成其他程序设计语言实现。
- 代码重构: 以更简明的形式改写,并修订了旧版的若干错漏之处。采用现代C++语言描述,程序编写与表达更为自然。
- 学以致用: 更加全面地讲解了STL中的容器功能,同时也尽量利用algorithm库的算法函数构建代码,从而增强本书的实用性。

## 组织

教材内容组织安排上遵循以抽象数据类型为核心的思想,突出如何以抽象数据类型搭建算法和应用程序。为此,本书中很多章节标示了分类:

- [使用] 主要讲解STL中容器的功能,解决抽象数据类型如何使用的问题。
- [技巧] 深入剖析算法与编程的技巧,主要关注效率的提升。
- [实例] 针对具体案例给出完整的解决方案,展示数据结构与算法的典型应用场景。

## 说明

- 本书尽量使用现代C++编写程序,因此请读者下载最新版本的编译器。不过考虑到编程习惯,我们没有完全使用现代C++,例如空指针`nullptr`。此外,本书中的部分代码还使用了模块(module)形式实现,不妨参阅DSAD项目。

- 除了若干单独执行的程序, 书中的其他代码可编译后并链接(例如在IDE中以工程形式组织并整合), 而包含main函数的源文件为DSAD.cpp(见附录), 可调用其中的演示程序查看运行效果, 这样使用本书的代码会更加方便.
- 初学者可以略过数学推导较多的部分, 特别是算法分析, 但得了解常见渐近记号的含义并能加以对比.
- 在一般情况下, 本书中 $\log(\cdot)$ 一般指代 $\log_2(\cdot)$ , 即我们主要讨论以2为底的对数. 此外, 这种表示方法意味着渐近记号中对数的底并无实质性影响.
- 不同字体代表不同对象, 另外如果同名则意味着指代相同: 例如 $n$ 表示数学公式中的符号, 而`n`代表程序中使用的变量, 不过它们都指代同一个量, 例如某个数组的长度.
- 本书仅讲解一些基本的排序和查找内容, 并分散于不同章节, 而更深入的相关内容留待算法课程中再讨论.
- 书中加星号的章节和习题为选学与选做内容, 读者可根据需要取舍.
- 为了突出C/C++语言下标的特色, 本书中的代码以编号标注并从0开始(自然数亦是如此). 另外, 单行代码或者算法运行步骤以 $\rightarrow$ 作为指示.
- 每章的注释部分提供了“代码延伸阅读”, 作为进阶学习之用.

## 致谢

在本书的写作过程中, 得到了诸多朋友的大力支持(初稿写作的致谢名单详见第1版), 特别是在微博以及GitHub上的网络互动带来的教学相长(恕我无法一一列出). 感谢本书编辑龙启铭先生对作者的耐心包容. 感谢我的父亲为本书题字. 特别还要感谢试用本书的各位同学, 正是他们促进了这本书的不断完善.

由于作者水平所限, 恳请大方之家对本书指正建言. 发现书中问题的读者朋友将获得一张签名电子明信片(以提交时间次序为准), 并会列入致谢名单. 在发送您的意见和建议之前, 请先查阅勘误表(<https://github.com/xiexiexx/DSAD>).

谢 颢

2021 年 6 月

于“算法时空”