

# 基于快速排序算法的综合实验教学案例设计\*

王正丽 张海宁 申国伟

贵州大学计算机科学与技术学院, 贵阳 550025

**摘要** 在授课课时减少的背景下, 如何通过设计综合型教学案例来训练学生算法分析与优化的能力是教师在教学过程中需要关注的教学环节。论文设计一个基于快速排序算法的综合实验教学案例, 通过对案例的分层递进讲解来启发学生探索算法优化过程。案例利用快速排序算法的编程实现多样、可优化空间高、方便设置大规模测试用例以及可视化效果展示直观等特点, 引导学生实现不同层次的算法优化目标, 结合 OJ 平台完成算法在不同规模数据下的性能验证。实践表明, 通过综合型教学案例的引入能够有效地吸引学生课堂注意力, 激励学生挑战不同难度的任务目标。

**关键字** 数据结构, 快速排序, 教学案例

## Design of Comprehensive Experimental Teaching Cases Based on QuickSort Algorithm

Zhengli Wang Haining Zhang Guowei Shen

College of Computer Science and Technology of GuiZhou University  
GuiYang 550025, China;  
wangzhengli916@163.com

**Abstract**—In the context of reduced class hours, how to train students' algorithm analysis and optimization skills through the design of integrated teaching cases is a key focus for educators. This paper presents a structured and progressive teaching case based on the quicksort algorithm to inspire students to explore the algorithm optimization process. The case leverages the characteristics of the quicksort algorithm, including diverse programming implementations, high potential for optimization, ease of setting up large-scale test cases, and intuitive visual effects. It guides students to achieve different levels of algorithm optimization goals and validates performance on various scales of data using online judge (OJ) platforms. Practical evidence shows that the introduction of integrated teaching cases can effectively capture students' attention in class and motivate them to tackle tasks of varying difficulty.

**Keywords**—Data Structure, QuickSort, Teaching Case

### 1 引言

随着教学课时的减少和《数据结构课程设计》教学环节的删减, 如何在有限的课时里训练和提升学生在数据结构选择、算法的分析与优化的综合能力, 是授课教师需要重点关注并解决的实际教学问题之一。数据的排序问题既向下关联数据的物理存储方式(顺序/链式), 又向上支持扩展应用(排名估计、热搜推荐等), 同时还方便设计大规模测试用例对算法进行验证等, 适合作为课程的综合教学案例设计来逐层拆解, 锻炼学生的算法分析、算法优化等综合能力。

《数据结构》课程中排序的教学内容主要包括插入排序、希尔排序、冒泡排序、快速排序、堆排序、归并排序、基数排序等排序算法。在众多排序算法中, 快速排序因其排序速度快而著称, 算法应用了分治思

想, 通过选取中心点来跟待排数据不断比较, 将其分割成以中心点为分界点的小序列, 逐层减小问题规模的方式来实现数据的快速排序, 最后将已完成的有序小序列到底向上地合并成整体有序的数据序列。快速排序的实现是变化多样的, 不同的编程实现下, 代码的执行效率差别很大, 例如: 在 out-of-place 和 in-place 两种算法策略下, 空间复杂度分别为  $O(n)$  和  $O(1)$ , 如何编程实现将空间复杂度从  $O(n)$  降到  $O(1)$ ? 不同的 pivot (中心点) 选取策略下, 性能可能从理论上的  $O(n \log n)$  下降到  $O(n^2)$ , 如何优化 pivot 的选取策略以提高算法效率? 等问题, 都可以通过教学设计来引导学生进行优化探索。笔者在唐麟等<sup>[1]</sup>的启发下, 针对快速排序算法变化多样的特点, 设计逐层递进式的综合实验教学案例并辅助实验验证<sup>[2]</sup>, 启发学生对算法进行深入思考并展开讨论, 帮助同学持续地、系统地理解分治思想和体会算法优化的过程, 达到融会贯通、举一反三的教学效果<sup>[3]</sup>, 加深学生对排序算法的理解, 同时也为以后的学习和实践打下良好基础。

\* 基金资助: 本文得到贵州大学校级课程思政示范项目 (kcsz2024073) 资助。

## 2 快速排序概述

“快速排序 (QuickSort)”是一种高效的排序算法,采用分治法的策略,将大问题分解为小问题解决<sup>[4]</sup>。其基本思想是:选择一个基准元素,通过一趟排序将待排记录分隔成独立两部分,其中一部分记录关键字均比另一部分的关键字小,然后分别对这两部分你继续进行快速排序。快速排序的时间复杂度是 $O(n\log n)$ 。双边快速排序算法是一种经典的快速排序算法,其单趟快速排序算法步骤如下。

输入:待排序列  $array[0,1,2,\dots,n-1]$ 。其中,序列中待排序列元素个数为  $n$ ,  $array$  中每个元素包含两个项:  $key$  和  $value$ ,  $key$  为排序主键,  $value$  为元素的内容。

Step1: 从  $array[]$  中选取第 1 个记录作为  $pivot$ , 设置双边指针  $low$  和  $high$  (初始值  $low = 0$ ,  $high = array.length - 1$ )。

Step2: 使用  $high$  指针从表尾向表头依次进行搜索,当  $low < high$  时,若  $array[high] < pivot$ , 则交换两条记录并固定  $high$  指针,执行 Step3 中  $low$  指针的对比移动,否则  $high$  指针向表头方向移动一个单元位置。

Step3: 使用  $low$  指针从表头向表尾依次进行搜索。当  $low < high$  时,若  $array[low] > pivot$ , 则交换两条记录,同时固定  $low$  指针,执行  $high$  指针的移动,否则  $low$  指针向表尾方向移动一个单元位置;

Step4: 重复 Step2 和 Step3, 直到  $low \geq high$  时, 结束循环。

输出: 完成单趟快速排序的数据序列  $array[]$ 。

完成上述单趟快速排序后,  $array[]$  中  $pivot$  记录将被移动到最终排位位置上, 并以此  $pivot$  为分界点, 将  $array[]$  分割成左右两边子序列, 在  $pivot$  左边的子序列  $key$  值将全部小于  $pivot$ , 在其右边的子序列  $key$  值将全部大于  $pivot$ 。分割左右子序列后, 再分别对两个子序列使用单趟快速排序, 直到每个子序列只包含一个记录, 完成分治算法中“分”的过程; 在分治算法“治”的过程中, 从最后一层拆分处对序列采用“左边子序列+ $pivot$ +右边子序列”的合并方式, 依次向上一层合并, 最终得到完整有序的排序结果。在  $pivot$  选取效果理想的状态下, 快速排序将在执行  $\log n$  次后结束, 得到有序的  $array[]$  输出序列。

## 3 实验案例设计

在常规教学安排中, 冒泡排序之后直接讲解教材中经典的双边快速排序, 这种方式虽然可以让学生快速过渡到更为复杂的排序算法学习中, 但由于缺乏递进层次, 会导致学生难以对快速排序过程充分理解,

只能机械地记忆每一趟排序的实现步骤, 不利于学生对算法的学习和掌握。

为了培养学生算法分析与优化的综合能力, 设计了一个递进式的教学方案和综合实验方案来对快速排序的基本思想和逐步优化过程进行讲解。案例基于快速排序算法框架, 引导学生实现空间复杂度从  $O(n)$  到  $O(1)$  的优化、分析如何选取  $pivot$  能够有效避免排序退化到时间复杂度为  $O(n^2)$  的情况、尝试快速排序算法是否可以跟其他排序结合使用, 分析效果及原因等。此外, 在教学案例讲解中增加了算法执行过程的可视化环节<sup>[5]</sup>, 能够看到排序算法执行时数据位置的变化, 展示不同算法下数据的比较和移动过程, 分析数据的移动规律并对比算法执行效率。下面将详细阐述快速排序的综合实验教学设计方案。

案例针对快速排序算法设计了“基本思想-算法优化-效果验证-排序过程可视化”4 个递进式引导启发环节。首先, 讲解快速排序算法的核心思想及算法流程, 引导学生初步实现算法编程。紧接着, 分析代码的空间复杂度  $S(n)$  和时间复杂度  $T(n)$ 。在未经过设计和优化下, 简单的算法编程存在较大的优化空间, 空间复杂度优化难度低, 可以引导学生实现将  $S(n)$  从  $O(n)$  降低到  $O(1)$ , 以增加学生信心并引起兴趣; 继续分析算法中  $pivot$  的不同选择策略对执行时间的影响, 通过列举极端输入 (待排序列基本有序) 的情况下, 时间复杂度快速退化, 引导学生优化  $pivot$  选择策略, 提升算法效率。然后, 综合型的实验案例除自身算法的优化外, 可以引导学生尝试不同算法的结合对算法进行创新, 训练学生的综合能力。在算法实现过程中, 通过平台设计不同规模 (小规模、中规模、大规模) 数据测试<sup>[6]</sup>, 让学生感受到算法优化的性能提升效果。最后, 增加算法实现过程的可视化, 让学生看到不同算法执行时的数据变化规律及对比算法性能, 通过在线实验平台提供实验动画展示的多样化教学手段, 可以不断增进教学质量, 充分调动学生的学习热情, 培养和提高学生的学习能力<sup>[7]</sup>。

### 3.1 讲解快速排序算法的核心思想

快速排序是一种基于分治思想的排序算法, 其核心在于“分而治之”<sup>[8]</sup>。给定一个待排序的数组  $A[]$ , 快速排序的“分”阶段是: 首先, 从数组  $A[]$  中选择一个元素作为  $pivot$ , 然后将  $A[]$  分割成两个独立的子集  $A1$  和  $A2$ 。其中,  $A1$  包含所有小于或等于  $pivot$  的元素, 即  $A1 = \{a \in A \mid a \leq pivot\}$ ; 而  $A2$  则包含所有大于  $pivot$  的元素, 即  $A2 = \{a \in A \mid a > pivot\}$ 。这样, 子集  $A1$  中所有元素的  $key$  都比子集  $A2$  中所有元素的  $key$  小。接下来, 对这两个子集分别递归地进行“分”的操作, 直到每个子集中只有一个元素为止。最后,

将这些内部已排序的子集逐层合并，最终得到一个有序序列  $B[]$ ，即“治”阶段。

### 3.2 引导学生提出进行优化算法

在案例中，将通过 5 个步骤逐层地讲解算法分析与优化的过程：

**Step1:** 当采用最简单的 out-of-place 算法实现 3.1 中的快速排序过程时，其空间消耗主要用于辅助排序过程中产生的比较和移动。虽然实现代码编程简单，但带来的空间消耗却是  $O(n)$  的空间复杂度，在此基础上只需对学生简单引导就能实现 in-place 算法，提高算法的空间复杂度到  $O(1)$ ，减少空间消耗，让同学们初步体会简单的改变就能带来良好的性能优化效果。

**Step2:** 中心点 pivot 的选取好坏有时能直接影响快速排序算法的时间复杂度，在待排序列基本有序的情况下，甚至能使时间复杂度从  $O(n \log n)$  退化到接近  $O(n^2)$ 。优化 pivot 的选取策略，可以有效避免这种极端情况下算法时间复杂度大幅退化的现象。通过引导学生思考提出合理的选值方案，例如：选择待排序列首位、利用随机函数进行随机选择、采用三数中值分割法等，指导学生用实验方式来验证分析与假设，训练学生“分析-优化-验证”的综合能力。

**Step3:** 尝试不同算法的结合来提高算法效率是科学研究中常用的方法之一。案例除关注算法本身的优化外，扩展学生思路，通过尝试与其他算法相结合（如插入排序算法、归并排序算法）并验证算法效率，在实验验证效果的基础上引导学生分析算法改进有效的原因，例如：在快速排序框架下，每一次的分治过程能将数据接近的拆分为  $n/2$  的数据量，当迭代到当前待排数据长度小于某个值（例如  $n \leq 3$ ）时，使用插入排序能提升算法效率；当排序序列基本有序时，归并排序效率较高，引导学生尝试快速排序与归并排序进行结合，验证算法执行效率并分析原因。

### 3.3 对比不同算法实现的效率，验证算法优化效果，设计大规模数据进行测试

数据结构课程中，排序实验可以直接使用 int 型数据进行比较排序，比较适合用来创建大规模测试用例给学生测试性能。在 OJ 平台搭建实验平台、设计实验任务，在实验测试中除小规模调试数据外，设计大规模数据（5 万、10 万、20 万）用于排序，让学生切实体验大规模数据下算法优化对性能提升的影响，增加学生对性能优化的直观理解<sup>[9]</sup>。

### 3.4 排序可视化

在教学设计中增加了排序效果对比展示环节，结合希冀教学平台中算法可视化展示效果，如图 1 所示。为学生直观展示在不同规模的数据下，不同算法实现

的动态执行过程。在可视化展示中，将待排数据模拟成柱状线条，不同算法实现下，可以直观展示数据的执行过程；还能对比同一测试数据下，不同算法实现的执行过程及效率。使学生直观感受算法执行速度的对比，加深学生对算法执行过程的印象和提升学习兴趣<sup>[8]</sup>。

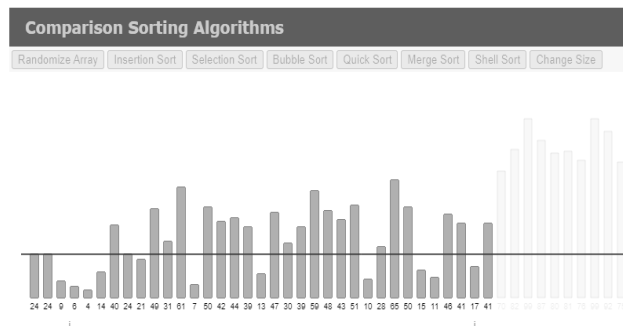


图 1 希冀平台中快速排序可视化过程

## 4 案例应用说明及教学效果对比分析

数据结构课程是计算机专业课程体系的重要组成部分。该课程主要面向本科二年级学生开设，学习基本数据结构，如逻辑结构、存储结构、算法设计与分析等相关理论，培养学生的模型抽象能力，为学生研究能力和创新能力的形成奠定基础<sup>[10]</sup>。

在教学过程中，授课教师首先结合动画课件分步骤讲解快速排序核心思想；通过引导学生探索、互动等方式让学生积极参与到优化算法的时间复杂度和空间复杂度的尝试中；在学生算法思想充分理解后，结合动态展示排序算法的执行过程，让学生分析数据在排序过程中位置的变化与算法执行的对应情况，加深对算法本身的理解。最后利用“实践教学平台”搭建在线实验环境，拆解本案例中 3.2 中所述实验方案来设计有层次、有梯度的实验任务目标，结合“闯关模式”帮助学生更好地认识自己对知识点的掌握水平及实践能力差异，从而能够更好地按照计划、循序渐进地在实践中不断学习和提高。

本案例在 2023-2024 学年第 1 学期软件工程 2203 班级得以应用。通过对比上一学年软件工程班学生对排序章节学习效果可知，综合案例的引入有助于提高选做综合性实验的学生占比数和提高章节小测的学生成绩，能够更好地调动学生的学习积极性，反映综合案例设计对提升教学效果的有效性。教学效果对比分析如下：

(1) 选做快速排序算法的比例有所提高。

在排序章节的众多排序算法中，由于学时的限制，实验任务并不要求学生对每一种排序都进行实现，通常是挑选其中 1 个完成即可。快速排序算法是实现难

度中上、有一定开发难度的算法,上一个教学周期中选择实现该算法的占比仅约为 18%,实现归并排序和堆排序的较多,且扩展中的要求大部分也均未实现。通过讲解综合性教学案例后,60%的同学选择尝试实现快速排序及进阶实验,并进行性能对比分析,如图 2 所示。通过选做结果表明综合性的讲解能够帮助学生更深入、更全面的理解算法本身,促进学生的尝试与探索。

选做快速排序算法实验比例明显提升

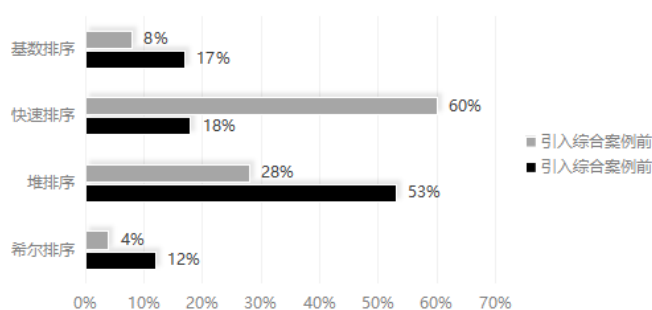


图 2 讲解综合案例前后选择实现快速排序算法的学生比例对比

(2) 章节小测成绩提升明显。

章节小测的不及格率从上一个教学周期的 22%降到了 7%,如图 3 所示。经过对教学过程进行分析,通过添加综合教学案例逐层递进式的讲解,吸引了同学更多的注意力,调动学生积极性,从而有效地提升了教学效果。

添加综合设计案例后章节小测不及格率下降

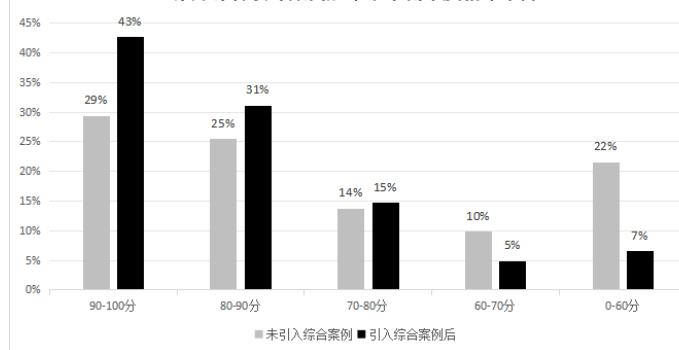


图 3 添加综合设计案例前后章节小测结果对比

(3) 学习积极性的提高。教师通过“掰开,揉碎”的逐层次深入式讲解,增加了学生的学习信心、提高了学生尝试算法优化的兴趣,更好地调动了学生的学习积极性,提高了课程活跃度。同时通过对一个算法的精讲,让学生理解并动手实现算法的分析与优化过程,从而达到训练其解决综合问题的能力。

## 5 结束语

《数据结构》的课程宗旨主要是要解决将数据合理地存入计算机,同时能够快速读取数据的问题,重点是培养学生的计算思维和解决实际问题的能力。在该课程各个章节中,查找章节和排序章节最适合用来对学生进行综合训练。教师通过对快速排序算法思路及执行过程进行详细讲解,进一步深化到算法时间性能优化、空间性能优化等方面的改进,启发学生对计算机科学进行讨论与探索,让学生在理解算法思想的同时,思考并实践逐步优化的过程,更好地训练和提升学生的计算思维和解决实际问题的能力。

## 参考文献

- [1] 唐麟, 张文辉. 数据结构实验教学过程的改进与实践[J]. 计算机教育, 2022(4):4.
- [2] 王玲芬, 李锡祚, 刘勇奎, 等. 在线评判在数据结构实践教学中的应用[J]. 计算机教育, 2016(4):71
- [3] 王彤, 鲍玉斌, 杨雷, 等. 数据结构实验金课建设教学探索与实践[J]. 实验室研究与探索, 2021, 40(5):9.
- [4] 严蔚敏, 李冬梅, 吴伟民. 数据结构(C语言版 第2版)[M]. 北京:人民邮电出版社, 2022.
- [5] 王燕, 罗佳琪, 王曙燕, 潘晓英, 曾艳, 白琳. 能力导向的数据结构课程“五环节”混合教学改革[J]. 计算机教育, 2023, 337(01):76-80
- [6] 杨武, 卢玲, 陈媛. 数据结构课程的大案例构建方法[J]. 计算机教育, 2018(7):4.
- [7] 徐金东, 阎维青. 计算机类课程贯通一体化建设——以“数据结构、图像处理”为例[J]. 计算机技术与教育, 2023, (11)2:97-100
- [8] 李一达, 黄维通. 针对快速排序改进的一些思考及其实现[J]. 计算机教育, 2018(5):6.
- [9] 赵海燕, 杜丽娟, 刘琨, 等. 高校课程思政教学实践探究——以“基于数据结构编程训练”课程为例[J]. 科教导刊, 2021(3):116-117.
- [10] 王兵书, 冯喜康, 马春燕. 文化元素融入“数据结构”课程思政的教学探索[J]. 计算机技术与教育, 2022, 10(3):60-64