



南京大學

# 本 科 毕 业 论 文

院 系 \_\_\_\_\_ 计算机科学与技术 \_\_\_\_\_

专 业 \_\_\_\_\_ 计算机科学与技术 \_\_\_\_\_

题 目 \_\_\_\_\_ 《算法设计与分析》课程实验设计 \_\_\_\_\_

年 级 \_\_\_\_\_ 2010 级 \_\_\_\_\_ 学 号 \_\_\_\_\_ 101220171 \_\_\_\_\_

学生姓名 \_\_\_\_\_ 周致远 \_\_\_\_\_

指导老师 \_\_\_\_\_ 黄宇 \_\_\_\_\_ 职 称 \_\_\_\_\_ 副教授 \_\_\_\_\_

论文提交日期 \_\_\_\_\_ 2014 年 6 月 8 日 \_\_\_\_\_

学 号 : 101220171

论文答辩日期 : \_\_\_\_年\_\_月\_\_日

指 导 教 师 : \_\_\_\_\_( 签字 )

# **Design and Implementation of Algorithm Course Labs**

by

**ZhiyuanZhou**

Directed by

**Yu Huang**

A thesis

Submitted to the Department of Computer Science and Technology

and the Graduate School

of Nanjing University

in Partial Fulfillment of the Requirements

for the degree of

**Bachelor of Engineering.**

Department of Computer Science and Technology

Nanjing University

June 8, 2014

# 南京大学本科生毕业论文中文摘要

毕业论文题目: 《算法设计与分析》课程实验设计  
计算机科学与技术 院系 计算机科学与技术 专业 2010 级本科生  
姓名: 周致远 指导教师(姓名、职称): 黄宇 副教授

## 摘 要

在计算机科学中,算法是一个计算的具体步骤,包含了解决问题的全部智能。对算法的研究与探索是计算机科学中的重要内容。在计算机本科教学中,《算法设计与分析》课程也有着极其重要的地位。它既是学生进一步学习、研究计算机科学的基石,又与实践紧密相联,是学生编程解决问题时的重要工具与手段。然而,我系的《算法设计与分析》课程偏重理论教学,实践教学内容仍有欠缺。

为了解决上述问题,本文根据《算法设计与分析》课程的具体需求,为该课程设计了一组实验,并设计实现了一个在线实验平台对这些实验进行测试、评分。具体而言,本文分析了《算法设计与分析》课程对实验的需求并梳理了该课程内容的重点、难点,然后针对这些重点、难点设计实验。其次,在每个实验中,本文根据课程教学安排,设计实验流程,编写框架代码,撰写实验教案,并根据学生可能的算法实现精心设计了输入输出。而在实验平台的设计与实现中,本文设计实现了一个在线实验平台。该平台主要对实验正确性进行评判并对时间和空间复杂度进行度量。该实验平台具备通用性,可以方便地在其上部署新的实验进行扩展。

关键词: 算法实验设计; 实验平台; 排序算法; 查找算法; 动态规划

# 南京大学本科毕业论文英文摘要

THESIS: **Design and Implementation of Algorithm Course Labs**

DEPARTMENT: **Department of Computer Science and Technology**

SPECIALIZATION: **Computer Science and Technology**

UNDERGRADUATE: **ZhiyuanZhou**

MENTOR: **Yu Huang**

## **Abstract**

In computer science, an algorithm is a step-by-step procedure for calculations. It contains all the intelligence that is needed for solving the corresponded problems. Researches and explorations in this field are important components of computer science. Therefore, for undergraduates majoring in computer science, the Algorithm Design and Analysis course is also of great importance. It not only constructs the foundations for them to further their studies and researches, but also equips students with skills and technologies to solve real-world problems. However, the Algorithm Design and Analysis course in the Computer Science Department of Nanjing University was biased towards the theories and lacked labs for students to apply the theories.

In order to solve the above problems, this paper designs a set of algorithm labs and implements an online lab platform for these labs. This platform is used to test the labs and give them score. To be more specific, this paper analyzes the Algorithm Design and Analysis course's demands for labs and accordingly designed the set of labs to meet the demands. Secondly, based on the course materials and teaching goals, this paper writes a lab guide, programs a framework and designs a set of reasonable inputs and outputs for each lab. Thirdly, this paper designs and implements an online judge system to evaluate the time complexity and space complexity of the labs. Due to the extensibility of this system, new labs can be deployed on it easily.

---

**Keywords:** algorithm lab design; lab platform; sorting algorithm; searching algorithm; dynamic programming

# 目 录

<b>第一章 引言</b>	<b>1</b>
1.1 背景 . . . . .	1
1.1.1 《算法设计与分析》课程简介 . . . . .	1
1.1.2 算法课程实验需求 . . . . .	4
1.2 本文贡献 . . . . .	5
1.2.1 实验设计 . . . . .	5
1.2.2 实验平台设计实现 . . . . .	5
1.3 本文组织 . . . . .	5
<b>第二章 相关工作</b>	<b>7</b>
2.1 算法实验设计 . . . . .	7
2.2 实验平台设计 . . . . .	7
2.3 重要算法 . . . . .	8
<b>第三章 实验设计</b>	<b>9</b>
3.1 实验一: 排序算法 . . . . .	10
3.1.1 实验概述 . . . . .	10
3.1.2 实验要求 . . . . .	11
3.1.3 设计思想 . . . . .	12
3.2 实验二: 电子词典设计 . . . . .	13
3.2.1 实验概述 . . . . .	13
3.2.2 实验要求 . . . . .	13
3.2.3 设计思想 . . . . .	14
3.3 实验三: N Puzzle 问题求解 . . . . .	14
3.3.1 实验概述 . . . . .	14
3.3.2 实验要求 . . . . .	16

3.3.3	设计思想 . . . . .	17
3.4	实验四: 六度分割理论验证 . . . . .	17
3.4.1	实验概述 . . . . .	17
3.4.2	实验要求 . . . . .	18
3.4.3	设计思想 . . . . .	19
3.5	实验五: 基于 Seam Carving 的图像压缩算法 . . . . .	19
3.5.1	实验概述 . . . . .	19
3.5.2	实验要求 . . . . .	22
3.5.3	设计思想 . . . . .	22
<b>第四章</b>	<b>实验平台设计</b>	<b>24</b>
4.1	设计概要 . . . . .	24
4.1.1	主要功能与处理流程 . . . . .	24
4.1.2	运行环境 . . . . .	25
4.2	正确性分析 . . . . .	25
4.3	时间复杂度分析 . . . . .	26
4.3.1	程序步数 . . . . .	26
4.3.2	关键程序步数 . . . . .	26
4.3.3	运行时间 . . . . .	27
4.3.4	CPU 分配时间 . . . . .	27
4.4	空间复杂度分析 . . . . .	27
<b>第五章</b>	<b>总结与展望</b>	<b>29</b>
5.1	总结 . . . . .	29
5.2	展望 . . . . .	29
	<b>参考文献</b>	<b>II</b>
	<b>致谢</b>	<b>III</b>



## 插 图

3.1	排序问题的可视化效果图 . . . . .	12
3.2	N Puzzle 问题示例 . . . . .	15
3.3	典型的 8 Puzzle 问题解决过程 . . . . .	15
3.4	六度分离理论事例 . . . . .	18
3.5	Seam Carving 算法使用前状态 . . . . .	20
3.6	Seam Carving 算法使用后状态 . . . . .	20
3.7	能量矩阵示意图 . . . . .	21
3.8	Seam 示意图 . . . . .	22
4.1	实验平台使用流程 . . . . .	24

## 第一章 引言

在计算机科学中，算法是一个计算的具体步骤，包含了解决问题的全部智能。对算法的研究与探索是计算机科学中非常重要的内容。在计算机本科教学中，《算法设计与分析》课程也有着极其重要的地位。它既是学生进一步学习、研究计算机科学的基石，又与实践紧密相联，是学生编程解决问题时的重要工具与手段。然而，我系的《算法设计与分析》课程偏重理论教学，实践教学内容仍有欠缺。

本文的主要内容就是根据《算法设计与分析》课程的具体需求，为该课程设计实验，并设计实现了一个在线实验平台对这些实验进行测试、评分。具体而言，本文分析了《算法设计与分析》课程对实验的需求，梳理分析了课程的重点、难点，并针对这些重点、难点设计了一组实验。其次，在每个出题点中，本文根据课程教学安排，设计实验流程，编写框架代码，撰写实验教案，并根据学生可能的算法实现精心设计了输入输出。而在实验平台的设计与实现中，本文设计实现了一个在线实验平台。该平台主要对实验结果正确性进行评判，对时间和空间复杂度进行度量。该实验平台具备一定的通用性，可以方便地在其上部署新的实验从而加以扩展。

本章主要内容首先简要介绍了《算法设计与分析》课程的主要内容，分析了其对实验的需求，最后给出了本文的组织结构。

### 1.1 背景

#### 1.1.1 《算法设计与分析》课程简介

算法设计与分析是计算机科学与技术专业学生必须熟练掌握的核心技能之一，它是问题求解的基础，也为计算机专业学生提供了用以解决实际问题的的重要手段。对算法设计与分析相关知识的深入学习也可以锻炼学生的思维能力与习惯，为后续专业课程的学习奠定良好的基础。《算法设计与分析》作为一门面向本科高年级学生的专业基础课程，在前置课程《数据结构》的基础上，系统化地介绍了各类重要算法的算法思想、基本原理和运用场景，也着重讲授了分析这些算法所需要的性能指标与分析方法。该课程要求学生不仅能熟练掌握算

法的工作原理，而且能灵活运用算法设计技术，结合实际情况设计算法解决问题。

课程教学内容以算法设计和算法分析作为主线展开，围绕算法问题系统地以下几个方面的具体知识：

#### 1.1.1.1 基础知识

算法设计和分析的基础知识包括图灵机模型（Turing Machine），算法分析标准和复杂度的渐进分析（Asymptotic Growth Rates）。图灵机模型是本课程所讨论算法的计算模型，它忽略了现实情况下的硬件的条件限制，而理想化地假设本课程讨论的所有算法都运行在无限的图灵机上。深刻理解图灵机模型是学习图灵机模型上一切算法的基础。算法分析标准包括了正确性（Correctness），时间复杂性（Time Complexity），空间复杂性（Space Complexity），简洁性（Simplicity）和最优性（Optimality）。其中时间复杂性和空间复杂性分别指算法在执行过程中所需的工作量多少和所占用的内存大小，对时间、空间复杂度的分析一般有最坏情况分析和平均情况分析两种情况。本课程中，对算法进行正确性的证明和时间、空间复杂度的分析是组织课程内容的重要线索。复杂度的渐进分析是指在分析复杂度关于输入规模变化时，忽略在极限情况下对复杂度没有影响的项和系数而进行的分析，复杂度的渐进分析是算法分析过程中的重要方法。

#### 1.1.1.2 算法思想

该课程设计的重要算法思想包括：分治法（Divide and Conquer）、贪心算法（Greedy Algorithm）和动态规划（Dynamic Programming）。分治法是指将复杂问题分为相同或相似的子问题，再将子问题分成更小的子问题直到最后子问题可以简单求解，并合并子问题的解以得到原问题的解的算法设计思想。这个思想是许多高效算法的设计基础。贪心算法是指在问题求解时，总是执行当前情况下最好的选择以解决问题的算法思想。动态规划是在分治法的思想下记录子问题的解以避免重复计算从而降低时空复杂性的算法设计思想。动态规划的设计思想使得很多问题从指数时间可解变为多项式时间可解。

### 1.1.1.3 重要算法

对课程设计算法进行归纳梳理，本文认为《算法设计与分析》课程主要涉及了以下重要算法：

- Sorting:
  - QuickSort
  - MergeSort
  - HeapSort
- Selection:
  - Finding Kth Largest Key
- Searching:
  - Hashing
  - Binary Search Tree
  - Red-Black Tree
- Graph Traversal:
  - Breadth First Search
  - Depth First Search
  - (Strongly) Connected Components
  - Transpose Graph
  - Task Scheduling(Topological Order)
- Greedy Algorithm:
  - Prim's Minimum Spanning Tree Algorithm
  - Kruskal's Minimum Spanning Tree Algorithm
  - Single-source Shortest-Path Algorithm
- Dynamic Programming:
  - All-pairs Shortest Path
  - Matrix Multiplication Problem
  - Optimal Binary Search Tree
  - Separating Sequence of Words
  - Changing Coins Problem

- String Matching:
  - KMP Algorithm

本文针对上述重要算法，选择重点、难点，设计实验并实现。

#### 1.1.1.4 计算理论

本部分内容简要介绍了 NP 完全理论、NP 完全问题的判定问题和采用近似方法求解 NP 问题的基本思路。

### 1.1.2 算法课程实验需求

在本科教学中，《算法设计与分析》是一门重要的课程。该课程着重讲授了算法的基础知识，例如算法复杂度的度量、基本的算法基础与实现等。这些知识是学生进一步学习计算机知识，研究计算机相关问题的基石。在面向计算机系学生的工作招聘中，算法设计与分析的能力也常常是考察的重点。算法课程的重要性可见一斑。

然而，在实际的教学过程中，算法课程又是一门公认的有难度的课程，它对学习者的学习和理解能力都提出了很大的挑战。一些学习者在学习算法的时候，由于其相对艰深，所以往往倾向于只求记忆，不求理解。然而，这样的学习方法虽然可以勉强应付考试，但是难以打下坚实的算法根基。另一方面，由于算法步骤纷繁复杂，有时学习者对算法的理解会似是而非。他们自认为对算法的各个细节全然理解，但是事实上有所偏差却不自知。

针对这些问题，本文认为在算法课程的教学中引入实验环节将大有裨益。首先，算法实验可以增进学习者对算法学习的兴趣。在实验中，算法不再是书本里的抽象知识，而是可以加以灵活运用来解决实际问题的工具。通过学以致用，学习者对算法的兴趣和热情会大大增加。其次，学习者可以通过精心设计的实验加深对算法的理解。通过对算法一步步地进行编程实现，学习者有机会更清晰地了解算法每个细节。在使用算法进行问题求解的过程中，学习者也会对算法背后的思想精髓有更多的体会。再次，学习者也可以通过实验验证其对算法的理解。在实验里，学习者对算法理解的任何一点错误与模糊都会导致实验失败，因此，他们可以通过实验来暴露自己的错误理解并加以纠正。更次，

实验可以增加解决实际问题的能力。对于计算机专业的学生而言，动手编程解决问题的能力是极为重要的，算法实验可以进一步锻炼加强这一重要能力。最后，算法实验还是很好的进一步研究、探索算法的基石。在实现了解决问题的经典算法之后，学生有机会对算法加以改进或者完全重构，并通过实验讨论其优劣，从而在可能的情况下进行一定的创新尝试。

## 1.2 本文贡献

本论文在分析了《算法设计与分析》课程的实验需求后，设计、实现了的课程实验的两个重要组成部分：一组各有侧重且覆盖课程内容的算法实验和与之配套的实验平台。

### 1.2.1 实验设计

具体而言，本文基于课程知识点设计了一组实验，这组实验由以下五个具体实验构成：

- 实验一: 排序算法
- 实验二: 电子词典设计
- 实验三: N Puzzle 问题求解
- 实验四: 六度分割理论验证
- 实验五: 基于 Seam Carving 的图像压缩算法

这五个实验各自针对一类算法，并有着各自的教学目的。整体上，该组实验覆盖了教学内容，具有一定的完备性。

### 1.2.2 实验平台设计实现

本文设计实现了一个通用的算法实验平台，该平台展示了实验讲义与要求，并对算法实现进行正确性的评判以及时间、空间复杂度的测量。在一些实验中，实验平台也以竞赛或可视化的形式展现实验参与者的实验结果。

## 1.3 本文组织

本文后续内容组织如下：

第二章介绍了本文的相关工作。这些相关工作包括了前人对算法实验设计的探索，也包括了本文设计的算法实验中涉及的一些算法。

第三章介绍了文本设计的五个算法实验。每个实验都介绍了其实验概要、实验要求和设计思想等内容。

第四章介绍了本文设计实现的算法实验平台，着重介绍了主要功能和时空复杂度的度量方法。

第五章是总结与展望。总结部分对本文工作进行了归纳，展望部分则提出了扩展实验平台为其他课程提供实验的愿景。

## 第二章 相关工作

本文主要为《算法设计与分析》设计了课程实验以及与之配套的实验平台。围绕这个主题，已有前人做出了许多相关工作。这些相关工作包括算法实验的设计，在线测试平台的设计实现以及算法实验中涉及的重要算法。本章就这三个方面展开介绍。

### 2.1 算法实验设计

算法课程作为计算机专业的核心课程，有着很长的历史。很多计算机教学工作曾努力将实验引入算法教学的课堂。McCracken 和 Daniel D 曾为算法课程设计了三个实验 [1]，然而这三个实验简单直白，没有达到本文所追求的趣味性和实用性。Eddins Steven L 和 Orchard Michael T 将 MATLAB 和 C 语言用于图像处理课程实验 [2]，该实验设计虽然针对的是图像处理课程，但是设计思想为本文提供了很多借鉴。

### 2.2 实验平台设计

在实验平台部分，北京大学 Online Judge System [3] 首先为 ACM/ICPC 训练而设计实现了一套在线评测系统，进而将其运用于大学生程序设计类课程教学中。类似的，Kosowski Adrian 等的工作 [4] 也设计实现了在线评测系统，并总结四年里将该系统用于教学的经验。Petit Jordi 等设计的平台 [5] 不仅用于教学目的，并且对外开放，实验者可以使用 22 种编程语言参与 800 多道题的训练。然而，由于这些系统的竞赛背景，与本文实现的系统相比，它们和教学契合不够紧密。事实上，它们需要教学内容和实验内容主动适应在线测试平台。相反，本文实现的教学平台是专为教学设计的，更具体地说，是针对《算法设计与分析》课程实验需求设计的，因此，本文实现的平台与该课程更契合。此外，在平台设计时，本文也注重以生动趣味的方式，如可视化效果等，来展现实验结果，这也是上述平台所不具备的。



## 2.3 重要算法

在实验的内容的设计上，许多重要算法为本文的实验设计提供了素材。实验一主要专注于多种排序算法 [6]，实验二主要涉及了查找算法 [6]，实验三扩展了课程内容，使用 A\* 算法 [7] 解决 8 Puzzle 问题 [8]，实验四使用了 Dijkstra 算法 [6] 来验证六度空间理论 [9]，实验五实现了 Seam Carving 图像压缩算法 [10]，考查了实验参与者对动态规划 [6] 的理解。

### 第三章 实验设计

在综合分析了《算法设计与分析》课程的主要内容以后，本文将算法课程主要内容分为三大部分：第一、算法设计与分析的重要指标和基本方法。这一部分主要涉及了算法的复杂性分析、复杂度的渐进表示等相关内容。第二，重要算法和其设计思想。这一部分内容包括了重要的排序算法、查找算法、图相关算法、贪心算法、动态规划、字符串匹配等。第三，算法理论。《算法设计与分析》课程讲授的算法理论包括 P 问题、NP 问题的定义与界定等。其中，由于算法理论相对抽象，本课程也由于是本科生课程，介绍并不深入，因此，实验设计涉及算法理论的部分相对较少。算法分析与设计的基本概念和具体的重要算法是本实验设计关注的重点。

更具体的来说，本文设计了五个实验：

- 实验一: 排序算法
- 实验二: 电子词典设计
- 实验三: N Puzzle 问题求解
- 实验四: 六度分割理论验证
- 实验五: 基于 Seam Carving 的图像压缩算法

其中，实验一要求实验参与者实现复杂度分别为  $O(n^2)$  和  $O(n \log n)$  的四种排序算法。该实验不仅要求实验参与者深刻理解多种排序算法（Sorting Algorithm）[6] 的原理与实现，也展示了在不同输入规模的下，完成相同任务的不同时间复杂度的算法所需关键程序步（Critical Operation）有巨大差异，让实验参与者直观体会大 O 表示下复杂度渐进函数的重要意义。因此，实验二在加深实验参与者对排序算法理解的同时，也着重关注了上述第一部分，既算法设计与分析的重要指标和基本方法。实验二要求实现一个电子词典（Electronic Word Book）功能。这个实验事实上是一个查找算法的现实应用，它考察了实验参与者对哈希算法、二分搜索算法、二叉搜索树、红黑树等相关算法的掌握。其次，通过精心设计算法输入，通过改变插入、删除、查找操作的比例、时机，本实验也强调了并不存在完美的算法，应该针对具体应用场景使用适宜的算法。实验三对课程中图遍历问题（Graph Traversal）中宽度优先搜索（Breadth First Search Skeleton）和深度优先搜索（Depth First Search Skeleton）[6] 进行了

扩展, 要求实验参与者使用搜索策略对以 8 Puzzle 为代表的 N Puzzle 问题 [8] 进行求解。该实验由于算法设计自由度较大, 不同设计的启发式方程 (Heuristic Function) [7] 对算法表现影响巨大, 因而, 在实验设计中以竞赛的模式展开, 以激发同学对算法实验的热情。实验四要求实验参与者根据 Facebook 上好友关系为基础, 设计算法以求出任意给定的用户距离用户集中最远用户的距离。这是对单源最短路径算法 (Single Source Shortest Path Algorithm) [6] 的具体应用。实验五将动态规划 (Dynamic Programming) 应用于 Shai Avidan 和 Ariel Shamir 与 2007 年提出的 Seam Carving 算法 [10], Seam Carving 是一种基于内容识别的图像压缩算法。实验四和实验五试图引导实验参与者将课堂中学习的算法应用于真实场景中来解决实际问题。本文设计的 5 个实验在实验目标上各有侧重, 符合《算法设计与分析》课程的各项教学目标的要求, 在内容上覆盖了该课程主要内容, 具有一定的完备性。下文将对五个实验做详尽介绍。

### 3.1 实验一: 排序算法

#### 3.1.1 实验概述

在计算机科学与数学中, 一个排序算法是一种能将一串数据依照特定排序方式的一种算法。排序算法是计算机科学中研究最广泛的算法之一, 许多分治算法都是排序算法的某种变形, 因此对排序算法深入研究很有必要。排序问题界定、描述简单直白, 许多算法直观且易于理解, 但又不缺乏深度, 初学者从排序算法着手开始算法学习是合理可行的。就本课程而言, 排序算法即在所有算法中有着重要的地位, 又可以清晰地解释算法设计与分析所需要的重要概念, 因此是本课程着重介绍的第一类算法。

基于上述考虑, 本文设计的第一个实验就是围绕排序算法的。我们要求实验参与者实现四种重要的排序算法: 插入排序 (Insertion Sort), 快速排序 (Quick Sort), 归并排序 (Merge Sort) 和堆排序 (Heap Sort) [6]。其中, 插入排序是一个  $O(n^2)$  的排序算法, 该算法逻辑简单且符合直觉, 首先要求实验参与者实现该算法可以帮助实验参与者快速加深对排序问题的理解, 也帮助实验参与者对实验平台快速上手。其次实现的快速排序, 归并排序和堆排序相对难度加大一些, 给实验参与者提出一些挑战。这三个算法做为平均时间复杂度

(Average-Case Cost) 为  $O(n \log n)$  的算法, 各有优劣。快速排序做为最广泛实用的算法之一, 在平均情况下有着非常优异的表现, 然而它在最坏情况下会退化为  $O(N^2)$  的算法。归并排序虽然在平均情况和最坏情况下时间复杂度都是  $O(n \log n)$ , 但是其系数较大, 相对其他  $O(n \log n)$  的算法效率不高。此外, 归并排序在归并时需要一个大小与输入规模相关的辅助数组, 因此归并排序不是一个原地算法 (In-place Algorithm)。堆排序是一个相对完美的排序算法, 平均情况和最坏情况下的时间复杂度都为  $O(n \log n)$ , 事实上, 堆排序对相同长度的数组比较次数是固定的, 与输入无关。堆排序的空间复杂度为  $O(1)$ , 也无可挑剔。堆排序仅存的问题是在平均情况复杂度上较快速排序稍差一些。

本实验要求实验参与者在实验框架代码提供的接口下实现这四种排序。实验平台将反馈实验参与者实现的四种排序正确与否, 并以不同规模的输入测试实验参与者提交的代码, 并获取该代码所使用关键程序步数与输入规模的变化关系, 以此作图直观展现四种排序算法的优劣。

### 3.1.2 实验要求

本实验环境为 Linux 下的 C 语言编程。首先, 要求实验参与者下载框架代码并阅读理解。sortFrame.c 是实验框架代码, 该实验框架包含随机数列发生、输入文件读取和输出文件写入等功能。student.c 是实验参与者要实现的文件, 实验参与者需要实现其中的:

- int insertionSort(int \* array,int arrayLen);
- int quickSort(int \* array,int arrayLen);
- int mergeSort(int \* array,int arrayLen);
- int heapSort(int \* array,int arrayLen);

其中每个函数都接受一个 int 数组, 并使用指定算法对该数组内元算进行排序, 如果排序成功则返回 1。值得注意的是, 为了清晰地界定算法分析中关键步的概念, 本实验中所有元素间的比较要求使用框架代码 compare.h 中声明的的 int compare(int,int); 实验完成并在本地测试通过后, 实验参与者需将 student.c 上传至实验平台使用 Online Judge 进行评分。

测试平台用输入规模为 5、10、20、40、80、160 的输入测试实验参与者提交的四种算法, 并将结果绘图, 直观展示四种算法在不同输入下所需的关键程

序步数。

实验在完成了 student 所有函数后还有额外奖励，要求以 0-99 构造插入排序、快速排序和归并排序的最坏输入，并上传至实验平台。文件名分别为 InsertionSortWorstInput.txt、QuickSortWorstInput.txt、MergeSortWorstInput.txt，格式与示例输入一致。

### 3.1.3 设计思想

本实验是本文的第一个实验，在设计时既要满足实验目的，又要让实验参与者尽快熟悉实验平台，因此简单地要求实验参与者实现四种重要的排序算法。然而，对这种四种排序算法进行深入分析也有着很重要的意义。因此本文设计的实验用不同规模的数据对算法进行测试并将关键程序步作图，形象直观的展现了几种算法的优劣。

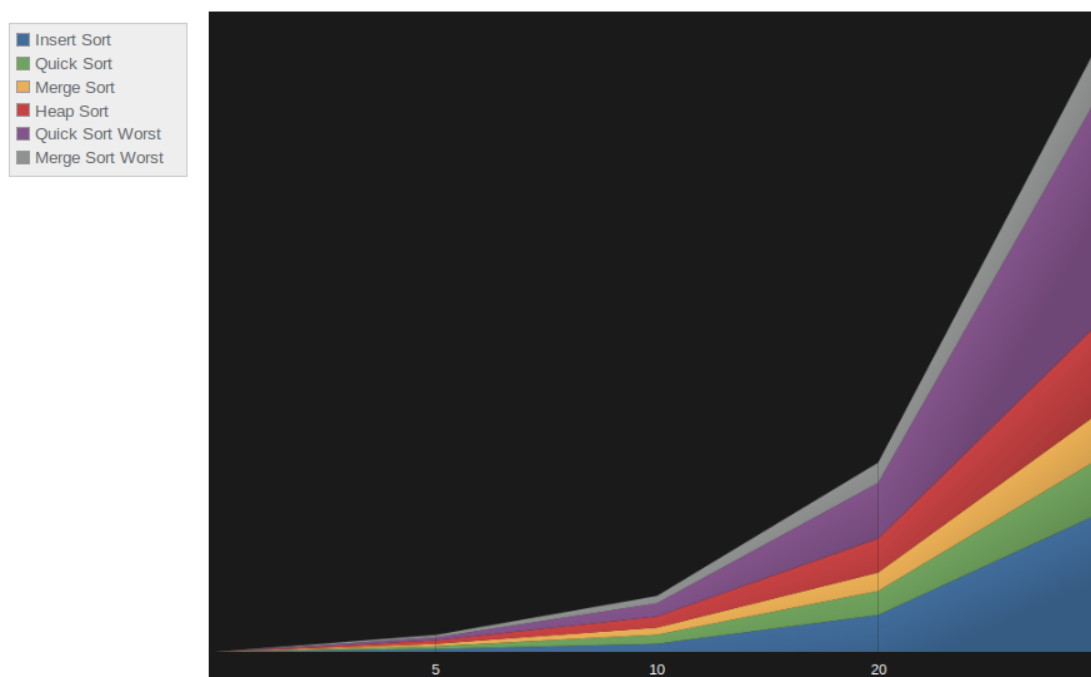


图 3.1 排序问题的可视化效果图

## 3.2 实验二: 电子词典设计

### 3.2.1 实验概述

本实验设计实现了一个单词本。该单词本的作用是帮助外语学习者记录生词, 并且能够让外语学习者方便地查询和复习。对于这样的生词本, 增、删、改、查是最基本的功能, 单词本还应提供对所有单词按字母序进行遍历的功能, 供外语学习者浏览记录的所有单词。除了这些基本功能外, 在本实验还要求该电子词典提供单词自动补全的功能, 即输入一个单词的开头, 即自动补全以该开头做为前缀的所有单词。

本实验平台维护的一个较大的单词集合, 并模拟用户操作对实验参与者实现的生词本进行上述增、删、改、查、补全和遍历操作。本实验要求实验参与者的代码能够对上述操作进行实时响应, 实验平台将对响应的正确性和响应速度进行评价, 并以此评价为实验结果评分。

### 3.2.2 实验要求

本实验环境为 Linux 下的 C 语言编程。首先要求实验参与者下载框架代码并阅读理解。ewdFrame.c 是实验框架代码, 该实验框架包含读写单词集、模拟用户操作等功能。student.c 是实验参与者要实现的文件, 实验参与者需要实现其中的:

- `int insert(struct word inputWord);`
- `int delete(struct word inputWord);`
- `struct word * search(char * wordName);`
- `struct word * fuzzySearch(char* wordName);`
- `int change(struct word inputWord);`
- `char* dicToString();`

其中, `insert`、`delete`、`change`、`search` 分别对应了增、删、改、查的功能, `fuzzysearch` 是单词补全功能, `dicToString` 将整个词典按字母序遍历输出。实验参与者要求完成上述函数。代码实现并在本地运行使用测时输入测试通过后, 上传 Online Judge 系统进行评分, Online Judge 将模拟用户输入, 反复调用上述函数, 并最后按照反应时间与内存使用评分。

### 3.2.3 设计思想

本实验主要考查了实验参与者对搜索算法的设计，事实上这些设计的优劣与增、删、改、查、补全、遍历这些操作的顺序和比例有关。在可能的设计中，如使用顺序表储存单词，并用折半搜索来查找，那么它查找时的时间复杂度是  $O(\log n)$ ，并不差，但是插入的时间复杂度  $O(n)$  并不太好。因此，如果单词本的使用者经常使用插入功能，而较少使用搜索功能，那单词本的性能表现就不好了。如使用二叉搜索树存储单词，则插入和查找的时间复杂度都是  $O(\log n)$ ，但值得注意的是，在最坏情况下，插入和查找的时间复杂度都会退化为  $O(n)$ 。因此，如果单词本的使用者恰好是一个从 A 向 Z 背单词的用户，其用户体验会很差。如果使用哈希表（Hash Table）来存储单词，固然增、删、改操作时间复杂度都接近  $O(1)$ ，但是哈希表也有两个显著的问题：第一，它占用空间较大，第二，对于哈希表而言，没有简易的方法进行遍历以及补全操作。因此，总的而言，在本实验中使用什么数据结构是一个开放性的问题，应该进行多方考虑。相对而言，红黑树和字典树是相对较优的数据结构。本实验设计希望实验参与者能通过该实验体会到各种查找算法在具体的问题中的优劣。

## 3.3 实验三: N Puzzle 问题求解

### 3.3.1 实验概述

以 8 puzzle 游戏 [8] 为代表的 N Puzzle 游戏一类常见的滑块游戏。它由一块有凹槽的板和写有数字的方块组成。以 8 Puzzle 为例，它在一个九宫格随机摆上写有 1-8 的数字的滑块，以及一个空位供滑块移动使用。游戏者需要水平移动滑块至空格处，通过若干次合法移动，使标有数字的滑块有序排列在目标位置上。



图 3.2 N Puzzle 问题示例

一个典型的 8 Puzzle 问题解决过程如下：

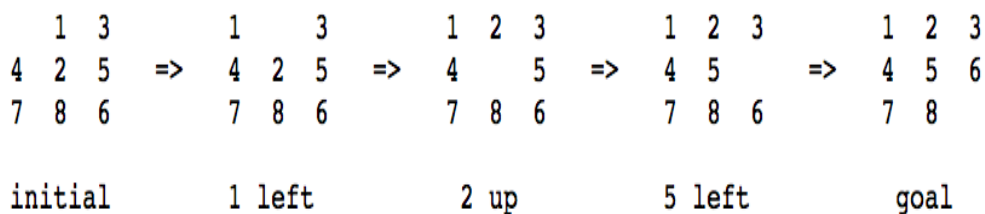


图 3.3 典型的 8 Puzzle 问题解决过程

对于  $2 \times 2$  方格上的 3 puzzle 问题，我们使用宽度优先算法即可轻易找到最优解。我们将滑块状态视为树的节点，则初始状态就是树根，每个状态通过合法移动形成的新的状态即其子节点。对这棵树使用宽度优先算法搜索目标节点，最终找到的由根到目标的节点就是问题的最优解。虽然在方格边长  $N = 2$  的情况下，3 puzzle 问题的解决相对容易，但是在  $N$  较大的情况下，寻找  $N \times N$  方格形成的  $(N \times N - 1)$  Puzzle 问题的最优解是一个 NP-hard 问题 [8]。然后，在  $N$  相对较小的情况下，我们可以使用 A\* 算法 [7]，对该类问题进行求解。

在这种算法下，我们对上述搜索树，并不是按照宽度优先的顺序进行搜索，而是按照一定标准选择最有可能处于最优路径中的节点进行展开。那么，我们



应该选择展开的节点具有什么样的特征呢？首先，我们希望它在搜索树中相对较浅，因为节点的深度即从初始状态到达该节点所需的步数。由于我们在寻找最有解，我们自然希望该步数越少越好。其次，我们希望它离目标较近，近指的是通过合法的移动到达目标所需的最少步数少。然而，在求出问题的最终解前，我们无法准确知道每个节点到目标所需的步数，不过幸运的是，我们可以估计这个值。

具体而言，我们首先定义一个数据结构来描述节点。

```
struct node{  
    board[N][N];  
    int level;  
    struct node* fatherNode;  
}
```

初始状态即树根。在本实验中，实验参与者的任务是构建一棵搜索树，然后按照一定标准选择叶节点进行展开从而搜索。选择应展开的叶节点的标准如下：该叶节点的层数加上该叶节点到目标距离的估值最小。在本实验的估值中，我们有两种常用的估值函数 [7]：

- 汉明距离方程（Hamming priority function）：在错误位置的方块数。
- 曼哈顿距离方程（Manhattan priority function）：每一个方块到目标位置的曼哈顿距离和。

值得说明的是，1、任何一个状态的汉明距离还有曼哈顿距离都不大于它到目标的真实步数，2、在 A\* 算法中使用任何满足这一性质的估值函数都能确保最终结果最优。在实验报告中提交对这两点的证明，可以获得额外的奖励。

### 3.3.2 实验要求

本实验环境为 Linux 下的 C 语言编程。要求实验参与者下载框架代码并结合注释阅读理解实验。PSframe.c 是实验框架代码，该框架代码已实现了 N puzzle 问题的生成，输入输出文件的读写等功能。student.c 是实验参与者要实现的文件，实验参与者要求实现其中的函数 struct node\* getResult(struct node\* startNode)，并将该文件上传至 Online Judge 进行评分。本实验将根据实验结果进行竞赛：我们设计了由易到难的一组输入，我们将反馈 20 秒内实验参与者提

交的实验代码可以解出的问题数，并以此排序。解出相同问题数的实验参与者，以提交时间先后为依据进行排序。

### 3.3.3 设计思想

本实验略有超出课程教学大纲，但实验内容是对教学内容的一种有益扩展和补充。在教学中，我们着重介绍过使用队列来实现宽度优先算法：即每次都从队列的队首获取元素进行访问，并将该元素的子女在队尾入队。在本实验中，本质上讲我们只是要求实验参与者将上述算法中的队列泛化为优先级队列，优先数由启发式方程决定。因此，本文希望通过这个实验向实验参与者传递如下理念：《算法设计与分析》课程中介绍的算法是我们设计研究算法以解决问题的基础，它们虽然都是经典算法，但是不是不可修改的，我们需要学会运用其思想，改变其中一些实现以适应实际问题。

## 3.4 实验四: 六度分割理论验证

### 3.4.1 实验概述

六度分离（Six Degrees of Separation）[9] 是一个著名的社会学概念，该概念认为社会中的任何两个人之间，至多只隔着六个人就可以相互联系。这个概念最初是由 Frigyes Karinthy 提出的，并因 John Guare 的同名话剧而广为人知。

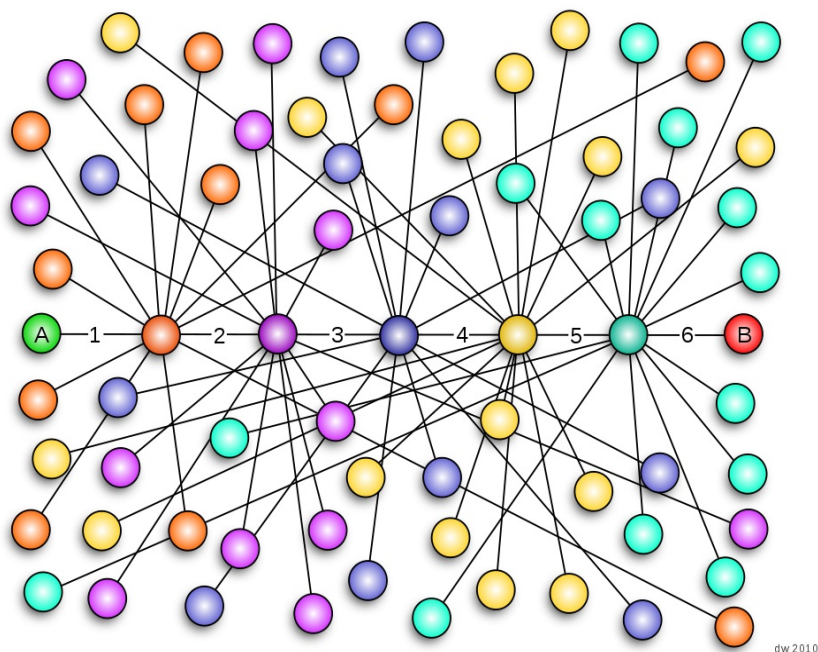


图 3.4 六度分离理论事例

由于早期技术条件限制，尽管 Stanley Milgram 等社会学家通过书信等方式对该理论进行了一定的研究 [11]，但大规模的研究始终无法进行。然而，随着互联网技术的发展与推广，对该理论的实验验证逐渐变为可能。2001 年，Duncan Watts 使用邮件系统重复了 Stanley Milgram 的实验 [12]。2007 年，Jure Leskovec 和 Eric Horvitz 使用了 Microsoft Messenger 的数据验证了六度分离理论，证明 Microsoft Messenger 用户之间的平均距离为 6 [13]。

本实验使用了 Facebook 好友关系作为数据，要求实验参与者使用单源最短路算法验证在该好友数据中，给定用户是否符合六度分割原理。更具体地说，本实验中每个 Facebook 用户都有一个唯一编号，每一条好友数据是一个  $\langle id1, id2 \rangle$  的二元组。实验参与者需要接受一个用户编号，并使用 Dijkstra 算法求出该用户到数据集中每一个用户最短距离的最大值。

### 3.4.2 实验要求

本实验环境为 Linux 下的 C 语言编程。首先，下载框架代码并结合注释阅读理解。sixDegreeFrame.c 是实验框架代码，该实验框架已完成了文件读写、好友关系建模的功能。student.c 是实验参与者要实现的文件，实验参与者需要实

现函数 `int maxDegree(int id)`，该函数接收一个用户 ID，需要返回该用户到数据集中所有用户距离的最大值（该值如小于 6 就说明六度分割对该用户成立）。实验完成并在本地测试通过后，实验参与者需将 `student.c` 上传至实验平台使用 Online Judge 进行评分。

### 3.4.3 设计思想

Dijkstra 算法是图路径中一个非常著名的算法，而以 Facebook 好友关系为代表的社会网络是典型的无向图，因此，在 Facebook 好友关系中使用 Dijkstra 算法是非常合适的。因此，本文使用真实数据在真实场景下设计了该实验，也希望借此引起实验参与者对社交网络相关领域的兴趣。

## 3.5 实验五: 基于 Seam Carving 的图像压缩算法

### 3.5.1 实验概述

Seam carving(接缝裁剪)是一个基于图像内容分析的图片智能裁剪算法 [10]，该算法每次将目标图片裁剪掉一个 Seam，从而使图片尺寸减一。在本算法中，一个竖直的 Seam 是一条连接图片第一行和最后一行的通路，该通路在每行中有且仅有一个像素，如下图中的红线。水平的 Seam 定义类似。下图展现了运用 Seam carving 算法裁剪了 150 个竖直 Seam 的结果图。



图 3.5 Seam Carving 算法使用前状态



图 3.6 Seam Carving 算法使用后状态

从这两张照片中可以清晰地看到，有别于普通裁剪或者缩放算法，Seam Carving 算法在缩小了原图 30% 的水平尺寸的同时，保存了原图的大部分细节（例如冲浪的人，云朵，小山等）。这个简单的而优雅的算法是 2007 年两位以色列教授 Shai Avidan 和 Ariel Shamir 首先提出的，如今已成为了图像处理软件，如 Adobe Photoshop 中的重要组成部分。具体而言，实现本算法需要实现一下三

个步骤:

### 一、计算能量矩阵

本实验使用了像素深度 24bit 的 BMP 格式图片作为算法实现对象。图片由像素矩阵构成，每个像素由 RGB 三个分量构成，每个分量的值在 0 到 255 之间。每个像素的能量取决于该像素与周边像素的颜色差别——差别越大则能量越高，也越难存在于将要删除到 Seam 中。最终由每个像素的能量组成的矩阵就是所求的能量矩阵，如下图:



图 3.7 能量矩阵示意图

### 二、选择要删的 Seam

下一步是选择一条 Seam，使它的所有像素能量和最小。要注意的是，在竖直 Seam 中，每个像素  $(x,y)$  向下相邻的像素只能是  $(x+1,y-1), (x+1,y), (x+1,y+1)$  中的一个。



图 3.8 Seam 示意图

### 三、删除选中 Seam

删除第二步中选中的 Seam，得到新的图片，该图片比原图在水平或者竖直方向上减少一个像素。反复对图片使用 Seam Carving 进行缩减，最终缩减至目标要求。

### 3.5.2 实验要求

本实验环境为 Linux 下的 C 语言编程。首先实验参与要要下载框架代码并解压缩。SCframe.c 是实验框架代码，该实验框架已完成了图片读写、能量矩阵的计算等功能，实验参与者需要阅读理解框架代码。student.c 是实验参与者要实现的文件，实验参与者需要实现函数在给定的能量矩阵中标定能量和最低的 Seam。实验完成并在本地测试通过后，实验参与者需将 student.c 上传至实验平台使用 Online Judge 进行评分。由于图片处理在最终结果上不要求完全一致，因此 Online Judge 系统会给出指向实验结果图片的链接，供下载查看。

### 3.5.3 设计思想

Seam Carving 算法在能量矩阵计算完成之后，要从能量矩阵中寻找一条能量和最低的 Seam 加以删除。寻找 Seam 有多种方法：最简单直白的方法是对图



进行暴力搜索，该算法的时间度为  $O(3^n)$ ，在图像稍大的情况下就不可能求出最优解了。因此，为实现本实验所要完成的实验目的，就要求实验参与者使用动态规划的思想设计算法，将时间复杂度降至  $O(n)$ 。

此外，本实验实现了一个实用功能的，这个功能已经成为新版 Photoshop 的一个特性。我们相信这个功能新颖有趣的实验对提高实验设计的趣味性大有裨益，也期望它能激发同学对算法学习的兴趣。



## 第四章 实验平台设计

在算法实验设计中，实验本身的设计固然重要，实验的呈现方式也应为实验参与者提供良好的用户体验，以增进实验效果。本文设计实现了一个在线的自动评价系统（Online Judge System）作为实验平台对实验参与者的实验结果进行评分。该实验平台提供了包括实验讲义、框架代码、测试用例在内的实验相关材料，也接受实验参与者提交的实验结果并进行反馈。我们相信使用实验平台与实验参与者交互，至少有以下优势：

- 24 小时实时提供实验反馈
- 节省批改评价实验结果所需的人力物力
- 对实验参与者得分管理省时省力
- 避免评价实验结果时的人为因素
- 算法输入不可见，因而可以有针对性地设计
- 可以使用多种反馈方式，增加实验趣味

具体地说，本文设计实现的实验平台有展示实验材料，并对实验结果正确性、时间复杂度和空间复杂度进行评判的功能。其中，为了满足实验设计的多样性，实验平台对正确性、时空复杂度的评价使用了多套机制。本章将对实验平台的设计展开详细介绍。

### 4.1 设计概要

#### 4.1.1 主要功能与处理流程

本实验平台以网站的形式展现，网站具有展示实验要求、实验结果、评价排名等功能，其主要处理流程如下：

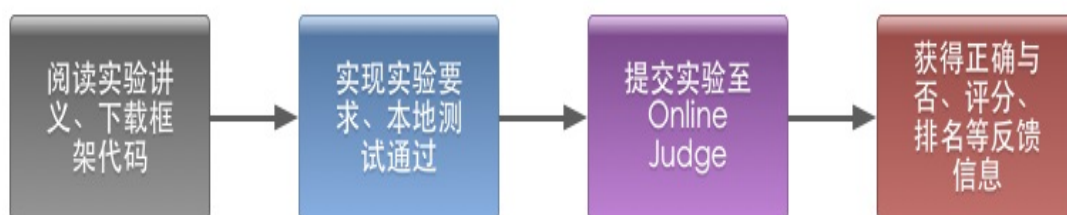


图 4.1 实验平台使用流程

用户首先访问实验网页，阅读实验要求并下载学生版本的框架代码。其次用户在实现了框架代码中所要求完成的函数，并在本地编译测试通过后，将实现的具体函数通过实验平台提交。实验平台接受用户代码，将其与测试版本的框架代码一同编译。如果编译成功则将设计好的、用户不可见的输入用例作为输入测试用户代码，然后将输出与正确答案相比对，并测算 CPU 时间、内存用量等数据。最终，实验平台将对用户结果进行评分，并将所有用户的评分结果以从高到低的顺序在网页上展示。

### 4.1.2 运行环境

本实验平台采用 LAMP 架构，既 Linux+Apache+Mysql+PHP 架构。Linux 系统是网站运行的操作系统，也是实验编译运行的操作系统。实验平台运行在 Apache version 2.2 服务器中，后台使用 PHP 语言，并用 Mysql 数据库中储存用户数据及实验结果。

具体实验要求使用 Linux 下使用 C 语言完成，并使用 gcc 进行编译。

## 4.2 正确性分析

正确性是对实验结果进行分析时所要考虑的重要指标。只有在实验结果是正确的前提条件下，对实验的时间、空间复杂度进行分析才有意义。

实验平台在默认情况下对实验结果正确性的评价是将实验输出与标准答案逐字比对，在二者完全一致的情况返回 true，反之返回 false。除此之外，本实验平台也支持具体实验按其实验要求，对正确性的评价方式进行扩展。设计者只需重写 `int if_result_right(FILE student_answer` 或者 `double if_result_right(FILE student_answer, ...)` 即可。前者通过返回 0 或 1 判定答案是否正确，后者通过返回一个 0 到 1 之间的 double 型数值反应答案在多大程度上正确的。实验平台在正确性评价方式上进行扩展主要是出于以下几点考虑：第一，很多问题的解并不一定是唯一的。例如，如果实验要求是找出“八皇后问题”的一个解的话，那么学生可能给出多种正确答案。因此，在这个问题里，`if_result_right()` 函数的功能就是判断学生给出的答案是否满足“八皇后问题”所要求的解的特性。第二，许多算法并不一定要找出最优解，而是要找出近似解。在这样的情况下，`if_result_right()` 函数就可以根据具体问题，对结果的好坏进行评判。对正确性评判机制

提供可以扩展的接口，使适用于本平台的算法实验灵活、多样，增进了平台的适应性。

### 4.3 时间复杂度分析

在计算机科学中，算法的时间复杂度是一个函数，它定量描述了该算法的运行时间。在算法实验中，对时间复杂度的度量是极其重要的，它直接体现了算法的效率。算法的时间渐进复杂度是一个关于算法的抽象概念，在算法的具体实验中，我们往往使用算法运行时间，或者程序步数对其进行分析。为了保证算法实验自由、灵活、多样，本实验平台提供以下四种对时间复杂度的度量方式。

#### 4.3.1 程序步数

程序步数是对时间复杂度的一个很直观的体现，在输入规模足够大的情况下，时间复杂度与程序步数是正相关的。此外，对程序步数进行统计还有简单直观的优点。具体的做法是，在测试版本的框架代码中加入一个全局的 `int step count`，并在学生提交代码的每一条语句后插装 “`step_count++;`”。最终在程序运行完毕之后，输出 `step_count` 作为结果。

#### 4.3.2 关键程序步数

按照时间复杂度的定义，一个算法的时间复杂度是关键程序步数关于输入规模的函数，因此，使用关键程序步数对时间复杂度进行度量是对正确的做法。在本实验平台中，我们提供了 `critical_operation()` 函数接口，供实验设计者根据具体算法实验自定义关键程序步，并要求学生在实现实验时，要使用关键程序步提供的操作时，必须调用该函数。最终，`critical_operation()` 函数的调用次数就是关键程序步数。举例而言，在基于比较的排序和查找算法中，该接口的实现就是 `int critical_operation(int a, int b)`，这个函数起到了比较的作用，当  $a < b$  时，返回 -1， $a == b$  时，返回 0， $a > b$  时返回 1。使用关键程序步数度量时间复杂度的一个潜在问题是，学生有在实验中作弊的机会。这种度量要求学生必须足够诚实，不自己另写代码实验避开调用 `critical_operation()` 函数。在未来工作中，本实验平台有可能在 `critical_operation()` 函数中引入 Client-Server 架构，使得 `critical`

operation() 函数返回值由远端服务器计算，并向用户隐藏计算逻辑，由此有可能规避这一弊端。

### 4.3.3 运行时间

减少程序运行时间是对算法时间复杂度进行分析研究最重要的目的，运行时间虽然机器相关，但是在相同的环境下，它与时间复杂度基本线性相关，因此，使用运行时间来度量时间复杂度是合理的。在实现这种度量时，需要学生代码调用前后，分别获取一次系统时间，并返回两者的时间差。这种度量的问题在于程序运行时间与机器相关，即使在同一台机器上进行测量，其结果也受机器负载影响，因此有一定的偶然性。

### 4.3.4 CPU 分配时间

CPU 分配时间是对运行时间测度的一种改良，它测量的是程序运行时真正分配到 CPU 时间片的时间，因此与机器负载关联相对较小。在 Linux 系统中，一个程序的 CPU 时间可以从/获得，一个程序占用的 CPU 时间既用户时间和内核时间相加。这种度量方法的劣势在于，它将框架代码耗时一并计入考虑之中，因此在框架代码完成功能较多，耗时较长的情况下不适用。

综上所述，这四种对时间复杂度的计量方法各有利弊，在设计实验时应该根据实验要求选择最适用的度量方式。

## 4.4 空间复杂度分析

空间复杂度是算法分析中的另一个重要组成部分，它讨论了算法在实现过程中占用的空间大小。在 Linux 系统下，对于程序占用内存情况可以通过访问/proc/PID/status 获得。在这个文件中，我们可以获得以下信息：

- VmSize: 进程当前使用虚拟内存的大小
- VmPeak: 进程使用虚拟内存大小的峰值
- VmRSS: 进程当前使用物理内存的大小
- VmHWM: 进程分配到物理内存大小的峰值
- VmData: 进程数据段的大小
- VmStk: 进程堆栈段的大小

- **VmExe**: 进程代码的大小

由于《算法设计与分析》课程所讨论的算法是基于的图灵机计算模型的抽象算法。因此我们实验中讨论的算法是与机器无关的，故而在对以上内存信息进行选择时，我们首先排除了与物理内存相关的信息。在虚拟内存使用情况的几项信息中，由于系统分配的内存可以手动 **free**，测量时当前的虚拟内存是没有意义的，因此我们认为，**VmPeak**，既虚拟内存使用峰值情况，是对算法空间复杂度的一个较好的度量。然而，进程运行时 **VmPeak** 的大小往往受到两个重要因素的影响：第一，算法的空间复杂度；第二，代码的内存管理能力。基于控制变量的思想，我们希望 **VmPeak** 能较好地体现空间复杂度，因此我们默认实验代码都具有良好的内存管理：这对我们的实验参与者提出了较高的要求。

## 第五章 总结与展望

《算法设计与分析》课程作为我系为高年级本科生开设的核心课程，对学生深入学习计算机科学起着非常重要的作用。它训练了学生的逻辑思维能力，提供了设计与分析算法的基本技能和手段，也介绍了许多重要算法，因此该课程为许多后续课程的学习打下了坚实的基础。然而，《算法设计与分析》课程是一门很有难度的课程，对学生的理解分析能力提出了很高的要求。为了配合课程的教学要求，帮助学生理解课程内容，本文为《算法设计与分析》课程设计了一组实验以及与其配套的实验平台。前文已对二者进行了仔细介绍，最后总结一下本文所做工作，并提出一些展望。

### 5.1 总结

本文首先介绍了《算法设计与分析》课程的相关背景，分析了该课程对算法实验的需求。然后梳理罗列了算法课程讲授的重要知识点，并以此为依据设计了一组实验，这组实验既各有侧重，又整体上覆盖了教学内容。最后，根据实验的需求，设计实现了承载本文实验的实验平台。更具体而言，本文主要做了以下工作：

- 对《算法设计与分析》课程实验进行了需求分析。
- 梳理梳理算法课程知识点，结合实际应用，设计实验。
- 撰写实验教程、框架代码、测试框架，设计输入输出。
- 模拟实现学生代码，根据可能的多种实现，改进实验设计。
- 设计实验平台，并使用 Linux、Apache、MySQL、PHP 架构实现。
- 设计机制对实验代码的正确性、时间复杂性、空间复杂性进行测试。

### 5.2 展望

本文已根据《算法设计与分析》课程初步设计了实验与实验平台。然而，本文设计的实验与实验平台都需要通过教学实践进行检验，并根据实际情况调整。另一方面，由于本文的实验平台具有一定的可扩展性，本文的设计思路也具有一定的借鉴意义，因此可以使用本文的已有成果为我系计算机科学的其他

课程设计实验。事实上，如果以本文工作为起点，设计一个为本系本科教学服务的实验体系，必定本科生教学大有裨益！

## 参考文献

- [1] McCracken and Daniel D. Three lab assignments” for an algorithms course. *ACM SIGCSE Bulletin*, 21(2):61–64, 1989.
- [2] Steven L Eddins and Michael T Orchard. Using matlab and c in an image processing lab course. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 515–519. IEEE, 1994.
- [3] Li Wen-xin and Guo Wei. Peking university online judge and its applications. *Journal of Changchun Post and Telecommunication Institute*, page S2, 2005.
- [4] Adrian Kosowski, Michał Małafiejski, and Tomasz Noiński. Application of an online judge & tester system in academic tuition. In *Advances in Web Based Learning–ICWL 2007*, pages 343–354. Springer, 2008.
- [5] Jordi Petit, Omer Giménez, and Salvador Roura. Jutge. org: an educational programming judge. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 445–450. ACM, 2012.
- [6] Sara Baase and Allen Van Gelder. *Computer Algorithms Introduction to Design and Analysis*. Pearson Education India, 2000.
- [7] David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.
- [8] Alexander Reinefeld. Complete solution of the eight-puzzle and the benefit of node ordering in ida\*. In *International Joint Conference on Artificial Intelligence*, pages 248–253, 1993.
- [9] Lei Zhang and Wanqing Tu. Six degrees of separation in online society. 2009.
- [10] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10, 2007.



- [11] Stanley Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.
- [12] Duncan J Watts. *Six degrees: The science of a connected age*. WW Norton & Company, 2004.
- [13] Jure Leskovec and Eric Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th international conference on World Wide Web*, pages 915–924. ACM, 2008.

## 致 谢

学长学姐对学弟学妹最好的馈赠，莫过于一份精心设计的实验。

我正是这样馈赠的受益者。在南京大学计算机系求学四年，我做过很多学长们精心设计的实验：蒋炎岩学长设计的《操作系统实验》，陈嘉学长留下的《编译原理实验》，陈健老师（学长）设计的《网络实验》……这些实验让我度过了许多个艰苦的夜晚，也构成了我大学四年努力求学的美好回忆。其间收获，一言难尽！

偶有一次听说蒋炎岩学长多年坚持奋战在《操作系统实验》教学一线，也是深受尹一通老师（学长）爱校爱系的精神鼓舞。忽然觉得“把南京大学建设成世界一流大学”不是一句口号，而是一个一届届学生都要为之奋斗的目标。就这样我萌生了试着设计一个实验的想法。感谢黄宇老师对我的大力支持和悉心指导，我终于在毕业设计里把这个想法变成了现实！

虽然由于本人水平有限，本文设计的算法实验与学长们的馈赠有着巨大的差距，但无论如何，这份设计表达了我对母校母系的一片心意！由衷感谢这四年来南京大学计算机系给我带来的一切，也祝愿母校母系越办越好！