

# 本科毕业论文（设计）开题报告

毕业论文（设计）题目		可视化算法交互式平台的设计与实现			
选题类型	应用型		课题来源	自选项目	
学 院	信息工程学院		专 业	计算机科学与技术	
导 师	潘勇浩		职 称	副教授	
姓 名	罗义	年 级	2018	学 号	201803666

## 1 立题依据

### 1.1 项目背景

算法是计算机科学与技术专业以及相关专业的核心课程之一，学好它也计算机专业后续的深入学习的基础<sup>[1]</sup>。然而算法的学习是一个十分抽象的过程，需要学生理解算法的运行机理，目前对于算法教学领域普遍采用 PPT 讲授的形式<sup>[2]</sup>，虽然有演示但只是执行程序展示结果，忽略了计算机系统内部算法数据的演示和算法运行的动态过程，学生无法和演示过程进行交互，影响学生对于算法本质的理解<sup>[3]</sup>，可以简单的演示算法的一些工作流程，但是仅仅是通过幻灯片和图片演示没有提供和学生交流互动的平台，很难让学生参与其中实际操作，降低了学生的学习兴趣，同时多媒体课件演示算法由于制作水平有限，好多算法演示的多媒体课件主要以展示知识为主，教师只能按照设定的演示顺序进行教学，这也不利于算法的教学，学生看到多媒体动画根本没时间细细思索，而动态可视化比如计算机软件、动画和交互式技术很少使用到，这一定程度上降低了多媒体教学的可视化效果，难以有效的展示数据结构和算法的抽象性和复杂性<sup>[4]</sup>。因此对于在教学中引入更加方便、易操作、扩展性好的算法可视化平台等研究都是有重大意义的。

### 1.2 国内外研究进展

国内的可视化程序的教学起步较晚，算法可视化程序也是较少的，大多数还是处在理论研究阶段。目前国内的算法可视化的教学研究基本上可以归纳为：①动画演示，即采用播放视频或者幻灯片的形式来演示算法的工作过程；②可视化调试，即根据编写好的程序代码，采用调试程序来理解算法的思想。起初在国内做的比较好的有应用到教学中的是清华大学严蔚敏教授 P 编著配有算法演示软件的《数据结构》系列教材中的配套可视化软件，该软件有 PASCAL 和 C 语言两种版本的解释，实现了数据结构书籍中的大部分典型算法的演示，将抽象的数据结构算法通过图形的方式展现出来，并且在演示的过程中可以查看配套语言的中间变量的数据变化，随后虽然关于算法与数据结构的书籍

逐渐增多，也出来了一些类似的可视化软件，但是仍然没有太大改善，绝大多数也停留于“展示执行过程与结果”的阶段。

国外对于算法可视化的研究比较早，也有很多优秀的可视化工具<sup>[5]</sup>。国外算法可视化系统大概有如下特点：①他们让学生输入数据，或者提供一个预先准备好的数据集；②他们能跟踪算法在数据集上的工作流程或动画的形式播放或分步演示。③随着算法的运行，数据的视觉表示可以被修改和更新。④一些系统还提供伪代码，让学生看每一行代码是如何作用于数据，并提供了机制系统的一些历史记录，学生可以复习以前的实施步骤。国外使用最广的开发算法可视化的方法是通过脚本语言模拟算法并且产生可视化效果<sup>[6]</sup>，比如 VisuAlgo、sorting-algorithms、illustrated-algorithms、AppletDemonstrationSite、VisualizingAlgorithms 等可视化。

### 1.3 设计目的与意义

之前的许多可视化算法平台，大多只是完成了“可视化”的功能，仅仅展示了对应算法的执行过程与结果，甚至执行的代码都不可见，这样的平台对于算法的用户而言，只看到了“开始”与“结果”，未能参与真正的“学习算法的过程”<sup>[7]</sup>，这样也就不完全满足辅助教学的需求<sup>[8]</sup>。所以，可视化算法交互式平台没有简简单单停留在可视化算法的展示的层面上，而是在前人的基础上，设计与实现更为复杂的“交互式”功能，例如，可视化算法交互式平台不仅可以让算法的用户看到对应的执行过程以及结果，还能引入《算法导论》等著名书籍作为资料库，将资料库纳入算法库中，方便用户在线查阅与学习，也可以支持多种常见的语言，让用户将精力主要投入于“算法思维学习”而避免将过多时间耗费于语言语法，甚至能在线编译代码，让用户亲身感受执行过程。同时，可视化算法交互式平台更进一步地侧重于“交互”<sup>[9]</sup>，这不仅仅可以让用户更快更深更准确地掌握对应的算法思维，也能更好地培养用户的代码能力。

## 2 主要内容及预期目标

### 2.1 研究主要内容

可视化算法交互式平台的核心组成为四部分，分别是算法库、服务处理器、跟踪器和算法可视化器，这四部分的具体功能如下：

(1) 算法库：算法库由算法源码库和算法资料库组成，算法源码库包含着整个平台已经完成渲染的算法源代码，每一种算法都对应着多种语言，用户可以从这里快速查看源码以及可视化结果；算法资料库包含着为用户提供的算法学习资料，这些资料选自《算法导论》《算法》等经典书籍。例如，用户可以在算法源码库中直接点击

“快速排序算法”，由于之前已经完成了渲染，所以不用等待过程，直接可以查看结果，用户也可以选择 Java、C++或者 JavaScript 对应的“快速排序算法”；用户可以在算法资料库中直接点击“快速排序算法”，那么面板中就会显示出关于该算法的文章，每一篇文章都对应着一个 markdown 文件，用户也可以自己修改与保存该文件。

(2) 服务处理器：平台所需要完成的功能都需要调用相应的 API，服务处理器的主要功能就是提供平台所需要的 API<sup>[11]</sup>，例如，用户写完了代码之后，点击“编译”，那么这个“编译”的命令就会首先进入服务处理器中，再由服务处理器结合“编译”对应的 API，打包交给跟踪器处理。

(3) 跟踪器：跟踪器中包含着可视化库，可视化库是用平台所支持的每种语言编写的。服务处理器提供了预渲染代码之后，跟踪器就将预渲染代码与可视化库结合，打包成可视化命令<sup>[12]</sup>，最后交给算法可视化器处理。

(4) 算法可视化器：算法可视化器最重要的功能是解释可视化命令来完成实际的算法可视化功能，解释命令的过程类似于模板引擎的解析，结合 Chart.js 库来渲染代码，算法可视化器也包含了丰富的 UI 组件，以便于提供丰富的样式，最后会结合 UI 组件把可视化命令渲染成对应的结果，再展示在面板中，以此呈现出最后的样式<sup>[10]</sup>。

(5) 上述四个核心部分的数据流程图如图 1 所示：

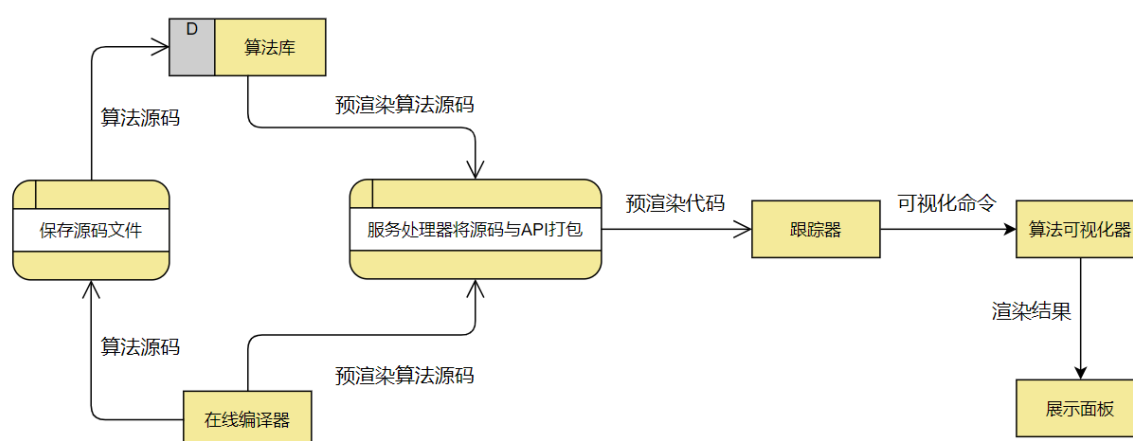


图 1 可视化算法交互式平台核心开发部分数据流程图

## 2.2 预期目标

(1) 在算法源码库中，实现大学课程中所涉及到的算法的可视化渲染，实现可支持的语言为 Java、C++、JavaScript。

(2) 在算法资料库中, 实现 markdown 语法的解析, 实现 markdown 文件的在线预览。

(3) 在编辑器中, 实现代码高亮功能, 实现代码自动补全功能<sup>[13]</sup>。

(4) 在控制台中, 实现代码的在线编译。

(5) 在可视化面板中, 实现可控渲染。例如, 对某一算法进行可视化渲染时, 用户可以对渲染过程进行暂停、增速、减速, 也可以跳转到某一具体的时间点, 查看该时间点的渲染结果。

(6) 在可视化面板中, 实现对生成的图像的基本移动与放缩, 以及点的移动、颜色对比渲染、数据标注等功能。例如, 对 Dijkstra 算法执行可视化渲染时, 路径渲染有明显的颜色对比, 路径距离有对应标注, 图像可放缩, 图像以及所有的点都可以拖动。

### 3 设计实施方案

#### 3.1 分析与设计

(1) 开发环境: 阿里云服务器, Chrome86, MySQL8.0

(2) 开发工具: VSCode

(3) 开发语言: JavaScript (TypeScript)

(4) 技术框架: 使用 react、react-router、react-redux 等作为基础构建技术栈, 使用 axios 做异步通信, 使用 node 做服务端, 使用 chart.js 做交互图表, 使用 bluebird 来快速处理 promise 对象加速渲染, 使用 tslint 做代码规范与检查。

(5) 具体设计方法: 将所有的代码以及资料都存入 MySQL 数据库中, 使用算法源码库以及算法资源库, 以及在线编译时, 会先通过 axios 向服务处理器发送渲染请求, 之后, 服务处理器结合提供的 API 将可视化代码与算法源码一起打包后发送给跟踪器, 跟踪器会将最后的可视化命令发送给算法可视化器, 由算法可视化器最终完成可视化渲染, 并展示在可视化面板中, 最后将整个项目部署到阿里云服务器上。具体的开发架构图如图 2 所示:

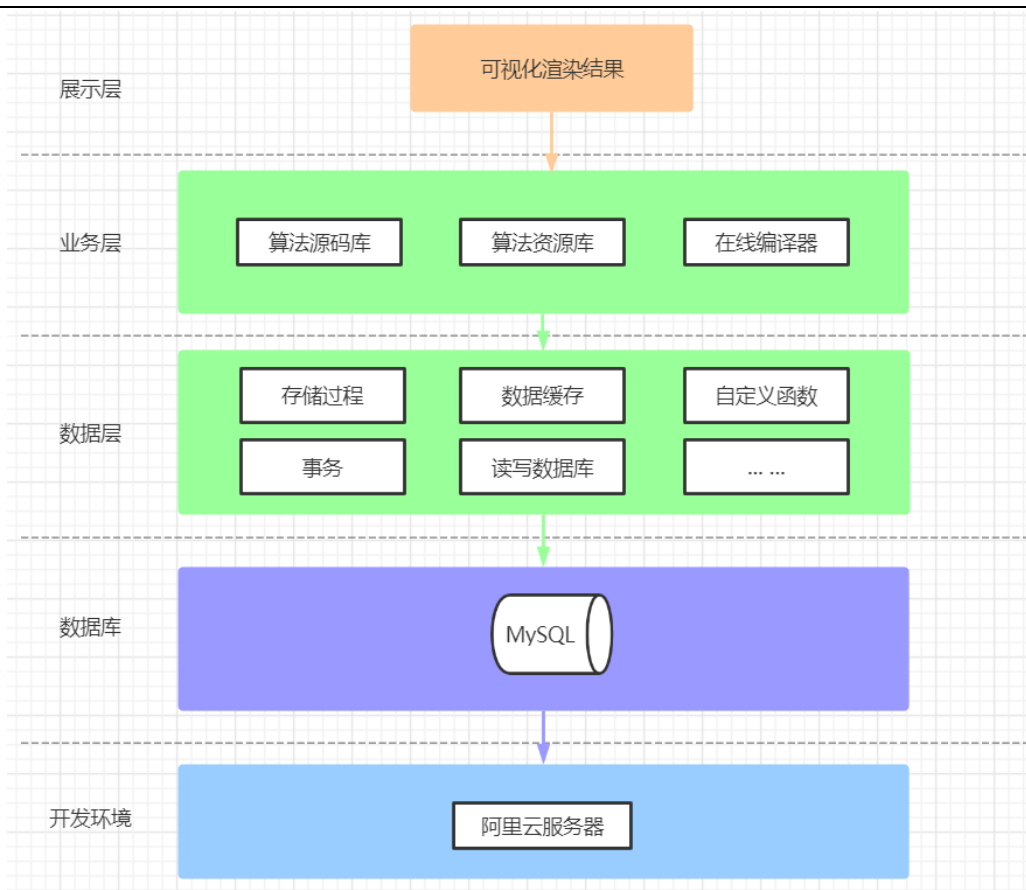


图 2 可视化算法交互式平台开发架构图

### 3.2 技术路线

(1) 对于算法库的开发过程，主要是通过 npm 提供的 fs 以及 mysql 模块，连通平台与数据库。

(2) 对于服务处理器的开发过程，主要是通过 npm 提供的 body-parser 模块抓取预处理的代码，再由 aws-sdk 模块实现存储，再通过 mocha 模块进行编译，最后通过 compression 中间件压缩文件，将压缩后的代码暴露给跟踪器部分。

(3) 对于跟踪器的开发过程，主要是通过几种语言对应的可视化库，抓取到可视化命令，从而将服务处理器抛出的代码进行预渲染。

(4) 对于算法可视化器的开发过程，主要是通过 markdown-it 解析 markdown 语法，完成在线预览，并且通过建立三元搜索树，实现代码自动补全，通过 highlight.js 实现代码高亮。

(5) 大致的技术路线如图 3 所示：

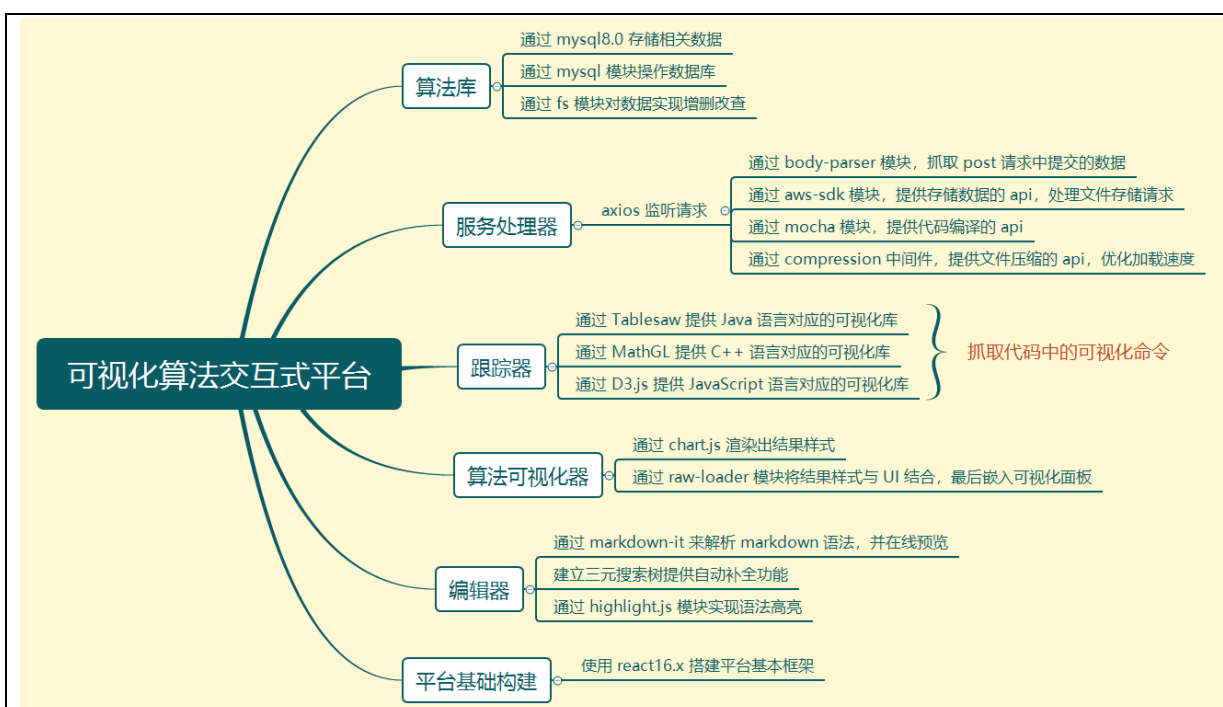


图 3 可视化算法交互式平台技术路线图

#### 4 研究进度安排

(1) 2021.04-2021.06: 学习算法可视化所需的基础知识，对应完成预期目标的第 1 部分的功能。

(2) 2021.06-2021.08: 进一步学习算法可视化，实现可控渲染，对应完成预期目标的第 5、6 部分的功能。

(3) 2021.08-2021.10: 学习制作在线编译器与编辑器，对应完成预期目标的第 2、3、4 部分的功能。

(4) 2021.10-2021.12: 搭建可视化算法交互式平台，并完成所有功能的集成测试。

(5) 2021.12-2022.02: 将项目部署到阿里云服务器中，并完成上线。

(6) 2022.02-2022.04: 完善毕业设计，开始撰写毕业论文，准备毕业论文答辩。

#### 参考文献

- [1] MU Yan-lei, GENG Yao-jun. 数据结构核心算法可视化系统设计与实现[J]. 现代计算机(专业版), 2019, 000(017):97-100.
- [2] MU Yan-lei, GENG Yao-jun. 数据结构核心算法可视化系统设计与实现[J]. 现代计算机(专业版), 2019, 000(017):97-100. 万萌. 基于深度学习的自然场景文字检测与识别方法研究[D]. 广东工业大学, 2019.
- [3] 严婷, 文欣秀, 赵嘉豪, 等. 基于 Python 的可视化数据分析平台设计与实现[J]. 计算机时代, 2017, 000(012):54-56.

- [4] 刘磊. 可视化数据挖掘方法与技术探究[J]. 通讯世界, 2017(6).
- [5] 杜鑫, 文双飞, 李科蓉. 算法可视化仿真教学辅助系统的设计——以数据结构课程算法为例[J]. 学园, 2018, v.11;No.297(36):84-85. 刘明英. 档案数字化过程中 OCR 技术的应用分析[J]. 中国高新技术企业, 2017, (5):55-56.
- [6] 胡瑞. 一种互联网大数据可视化分析系统:, CN109299168A[P]. 2019. [1]
- [7] 刘荣, 徐昕海, 刘呈,等. 基于虚拟现实技术的算法可视化实验研究——以最小生成树 Prim 算法为例[J]. 信息通信, 2020, 000(004):27-28. 徐新爱. 可视化教学研究与实践[J]. 教育研究前沿: 中英文版, 2017, 007(002):P.82-88.
- [8] 陈伟江. 基于 HTML5 的算法动画可视化平台[J]. 电子世界, 2017, 000(013):159.
- [9] 戴文鑫, 袁榕澳. 基于 HTML5 的算法可视化编辑器研究[J]. 信息与电脑(理论版), 2020, v.32;No.447(05):68-70.
- [10] 安智慧, 甘东, 陈威,等. 一种网页的编辑方法,系统及编辑器:, CN108255936A[P]. 2018.
- [11] 朱建. 超强通用编译器优化工具准确率是传统方法的 5 倍[J]. 计算机与网络, 2020, v.46;No.618(02):80-81.
- [12] 陈宏君, 张磊. 结构化文本语言编译器的虚拟机指令设计与优化[J]. 单片机与嵌入式系统应用, 2018, 018(005):23-27,48.
- [13] 杨博, 张能, 李善平,等. 智能代码补全研究综述[J]. 软件学报, 2020, 031(005):1435-1453.

导师意见:

导师签名:

年 月 日

注: 1. 选题类型: 基础型、应用基础型、应用型、调研型;

2. 课题来源: 国家级项目、省部级项目、横向合作项目、校级项目、自选项目。