



中山大學
SUN YAT-SEN UNIVERSITY

本科生实验报告

题目：广州地铁路线查询系统

姓名 温钰堂 武自厚

学号 20336012 20336013

院系 计算机院

专业 保密管理

指导老师 乔海燕

2021/12/30

目录

一、实验目的	2
二、实验环境	2
2.1 编程语言和开发工具	2
2.2 使用库.....	2
三、实验内容	3
四、分析与设计	3
4.1 需求分析.....	3
4.2 总体设计.....	5
4.3 服务端设计.....	6
4.4 前端设计.....	11
六、实验结果	12
结果总说.....	12
数据测试.....	12
结果展示.....	13
七、设计心得	15
Hanmur (温钰堂):.....	15
Naphthol (武自厚):.....	16
八、项目参考	17

一、实验目的

本实验面向数据结构与算法的初学者。

主要让实验者熟悉数据结构与算法，体验前后端分离开发流程。

二、实验环境

本实验基于 Golang、Visual Studio Code 平台开发，编程规范参考《Uber Go Style Guide》、PEP8编程规范、《阿里巴巴Java开发手册v1.4.0（详尽版）》，面向Windows、macOS、linux操作系统进行程序设计。

2.1 编程语言和开发工具

编程语言： Golang, Vue v3

开发工具： Visual Studio Code v1.61,

Goland Version: 2021.3.1

2.2 使用库

Golang:

标准库，无额外使用包

Vue v3:

axios库，AntDesign组件包

三、实验内容

通过本学期学习的数据结构与算法知识，编写一个基于Dijkstra算法的地铁路线查询程序。包括浏览器上的图形化显示和交互操作，完成最短时间查询和最少换乘查询。

四、分析与设计

4.1 需求分析

需求分析图

需求	简要说明	描述
数据	获取数据	用爬虫从地铁官网上读到数据库
	整合数据库	数据库数据格式化
前端	获取线路数据	从后端获取线路、站点等数据
	线路数据呈现	将数据填入选择框供用户选择
	输入处理	将输入的数据结构化，准备交由后端处理
	提交至后端	将数据输送至后端
	接收运算结果	接收来自后端的处理结果
	结果呈现	将处理结果以一定的形式呈现给用户

需求	简要说明	描述
服务端	数据库交互	从表格中提取目标信息
	获取地铁信息	获取地铁线路站点的全信息
	最短路径运算	计算起始点间最短路径
	最少换乘运算	计算起始点间最少换乘路径
	Json 包装	将获取数据包装成 json 文本
	架构 API	创建 API 服务

任务分配

Hanmur (温钰堂):

服务架构

Dijkstra 算法

项目整合

Naphthol (武自厚)

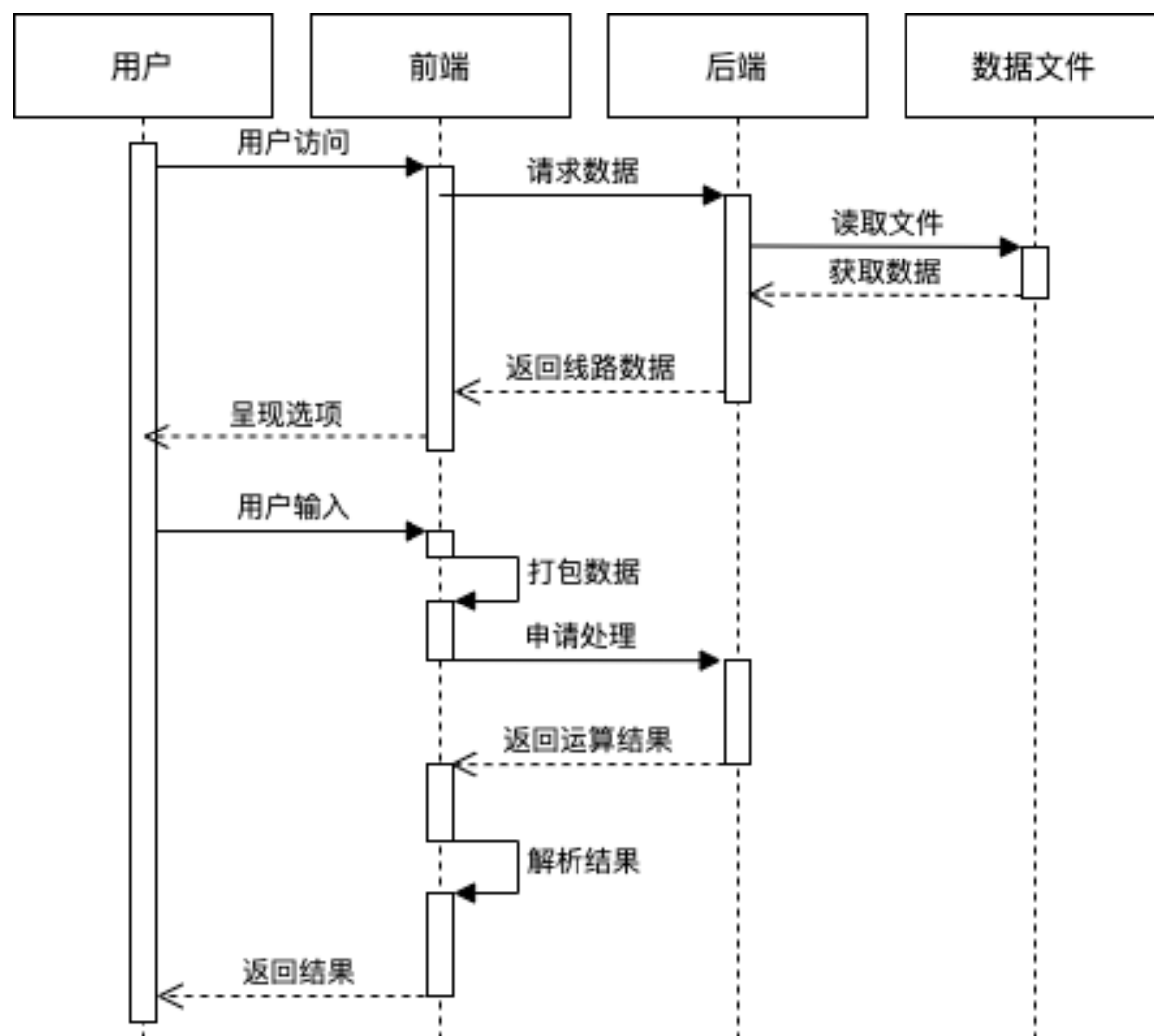
前端架构

爬虫读取地铁信息

项目整合

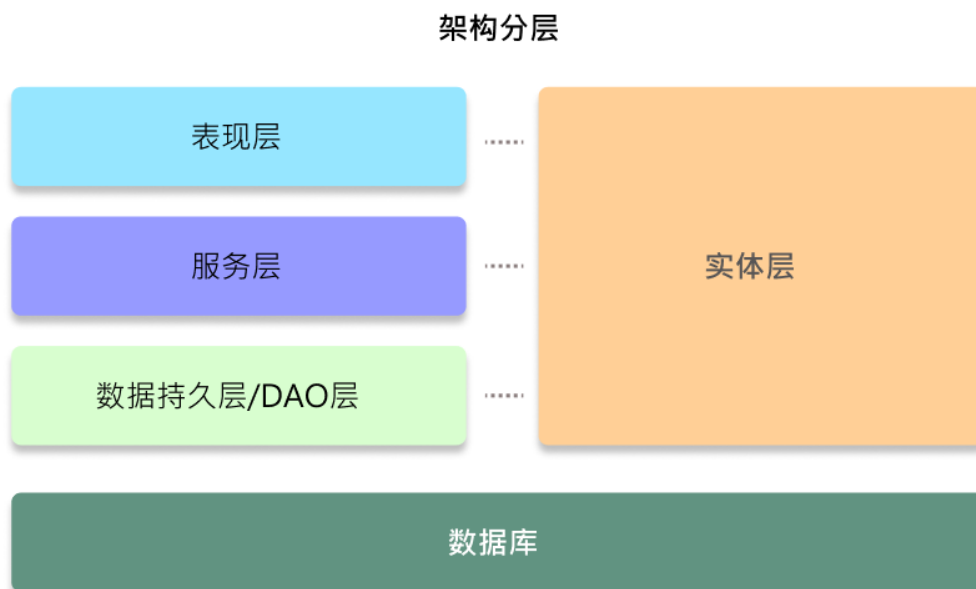
4.2 总体设计

运行流程图



4.3 服务端设计

应用架构图：



各层次设计

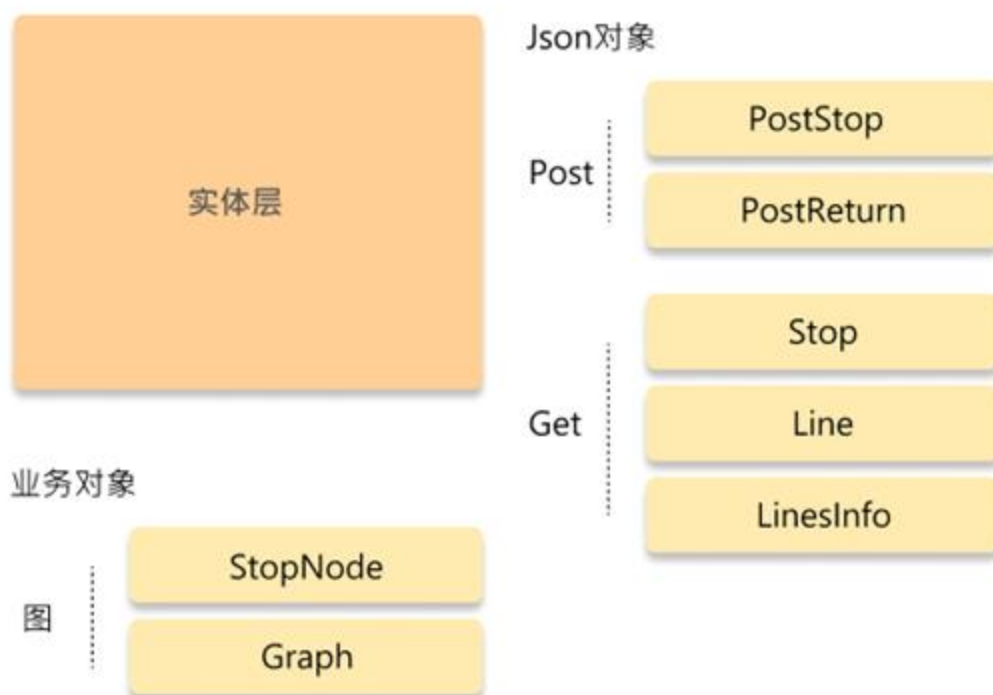
数据库



数据持久层/DAO 层



实体层



服务层



表现层



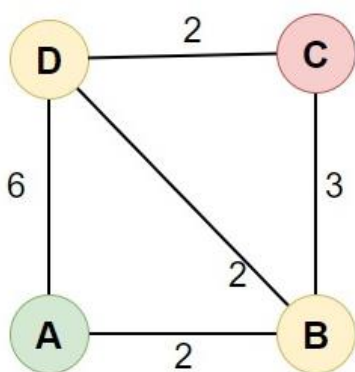
核心算法：Dijkstra

介绍：

每次从「未求出最短路径的点」中 **取出** 距离起点 **最小路径的点**，以这个点为桥梁 **刷新**「未求出最短路径的点」的距离，循环到所有点被获取

实现思路：

例图：



初始状态：（假设起点在 B 点）

将起点设置为 **已访问**，距离为 **0**，路径为 **[起点]**

操作点设置为 **起点**

操作点	B			
索引	A	B	C	D
Visited	0	1	0	0
Dis	MAX	0	MAX	MAX
Path	[]	[B]	[]	[]

循环直至所有点被访问(**visited** 均为 1):

(只演示第一次循环)

循环部分{

Step1 更新操作点相邻路径。若操作点相邻 **未访问点** 的路径距离少于 **Dis**

记录的距离, 更新 **Dis** 信息, 并替换 **相邻点 Path** 为 **操作点 Path**

操作点	B			
索引	A	B	C	D
Visited	0	1	0	0
Dis	2	0	3	2
Path	[B]	[B]	[B]	[B]

Step2, 更新操作点。选取 **未访问点** 中 **Dis** 最小的点, 其 **Path** 添加自身,

并更新操作点为该点, 更新其 **Visited** 为 1

操作点	A			
索引	A	B	C	D
Visited	1	1	0	0
Dis	2	0	3	2
Path	[B, A]	[B]	[B]	[B]

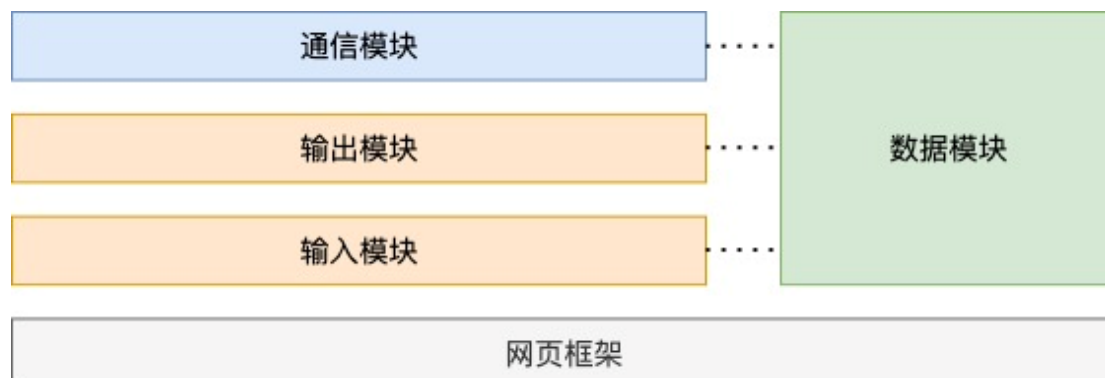
}

循环结束后, **path** 将记录起点到所有点的最短路径信息

注: 最短时间可以直接将时间代入为 **Dis**,

最少换乘可以设置换乘路线的 **Dis** 为 1, 其他路线的 **Dis** 为 0

4.4 前端设计



网页框架：vue-cli-3.0 以及 Ant Design-vue 3.0

通信模块：获取初始站点数据, 提交路线数据至后端并接收运算结果。

数据模块：负责存取、结构化、处理前端中的数据。

输入模块：呈现数据模块中的选项数据, 获取用户输入的路径数据。

输出模块：呈现后端运算结果。

六、实验结果

结果总说

实现了**时间最短查询**和**换乘最少查询**

完成了**浏览器前端**和**后端 API**

开发效率很高，有额外功能(鸡汤来喽)，结果准确无误

总的来说，本次开发以很让我们满意的状态落幕。

数据测试

起点	终点	查找类型	描述
一号线 广州东站	六号线 河沙	最短时间	<p>全程大概需要 62 分钟</p> <ul style="list-style-type: none">乘坐 一号线 西塱方向 3 站到 杨箕 113换乘 五号线 5乘坐 五号线 滘口方向 9 站到 坦尾 52换乘 六号线 6乘坐 六号线 浔峰岗方向 1 站到 河沙 64
三号线 大石	五号线 中山八	最少换乘	<p>全程需要换乘 1 次</p> <ul style="list-style-type: none">乘坐 三号线 机场北 (2 号航站楼) 方向 6 站到 珠江新城 310换乘 五号线 5乘坐 五号线 滘口方向 10 站到 中山八 53

结果展示

初始界面



选择起始点 (选择框进行选择)



点击“选好了” 进行查询，显示查询结果
(左右分别是时间最短及换乘最少结果)

请选择你的路线(左图单击可放大)

起点

终点

你的路线

厦滂 3 5 换乘 南村万博 7 6

☒ 时间最短 ☐ 换乘最少

全程大概需要17分钟

乘坐 **三号线** 番禺广场方向 2站到 汉溪长隆 3 3

换乘 **七号线** 7

乘坐 **七号线** 大学城南方方向 1站到 南村万博 7 6

 鸡汤来喽~
每次想省钱的时候，就是你智商到达顶峰之时！

请选择你的路线(左图单击可放大)

起点

终点

你的路线

厦滂 3 5 换乘 南村万博 7 6

☐ 时间最短 ☒ 换乘最少

全程需要换乘1次

乘坐 **三号线** 番禺广场方向 2站到 汉溪长隆 3 3

换乘 **七号线** 7

乘坐 **七号线** 大学城南方方向 1站到 南村万博 7 6

 鸡汤来喽~
每次想省钱的时候，就是你智商到达顶峰之时！

七、设计心得

Hanmur (温钰堂):

这次项目我们采用的是前后端分离的开发模式，这次我负责的是服务端 API 的编写。

本次开发的助力点:

1. 合理的项目架构

在开始开发前我进行了项目架构的设计，参考了阿里对 Golang 项目的设计架构。每个功能层次的问题只会出现在当前功能层次、完善的异常流动设计都帮助我快速排查出了写代码时遇到的问题。

2. 清晰的算法和数据形式

在提前设计好数据结构的情况下，算法对我来说只是把过程翻译成代码敲上去，效率很高也没有出现不可控的异常。

3. 高效的合作交流

和前端高效的合作交流，让我们出现的问题都被很快的解决，这点在大项目组上是很难实现的。

本次开发的痛点:

1. API 开发并不熟练，在跨域访问上纠结了相当一段时间。

2. 数据库操作并不熟练，本次开发大部分时间都在操作数据库上。

总的来说，本次开发我的收获是比较大的，对互联网开发有了更大的认识。

Naphthol (武自厚):

此次项目中我第一次接触了前后端分离的设计方式，这不可谓不是一个巨大的挑战。前端的开发与之前学习中熟知的命令行程序开发有着巨大的差异，W3C标准巨大的可定制性一度让我无所适从、手足无措。所幸以下几点帮助我成功啃下这个硬骨头：

1. 选择合适的框架。如果让我纯手打 Html+css+javascript 的话我估计一年都不会完成前端部分。由于 Vue.js 和 Ant Design Vue 使用，此次前端开发如虎添翼，使我不会拘泥于格式化的重复而是更专注于逻辑的呈现。
2. 随时准备学习，实践中学习。前端开发过程中出现了一些我无法理解的错误，直接放弃或者钻牛角尖取尝试并不是一个好的应对方式。我在遇到问题时善用了搜索引擎，对症下药地学习新知识并应用于错误修复。
3. 高效的沟通。项目开发不是单打独斗，错误、低效的沟通会严重拖累开发进度，比如此次开发中由于沟通问题造成了一些需求理解错误的情况。还好为时未晚，且这个失误发生之后我一直在和后端开发积极沟通、明确需求，在协作与交流中才会有 $1+1>2$ 的效果。

综上，我个人认为在此次大作业开发中，我学到了比其他人更多、更广的知识，并且积累了一定的开发经验。相信在未来的学习、开发中，这段经历将是我的“组件库”中一个重要的部分。

八、项目参考

Go语言http服务参考自：

[Go语言http-Request教程 | 🐼❤️🤖 \(mojotv.cn\)](#)

Go语言学习自：

[《The Go Programming Language》](#)

[分类: Go语言 | Hanmur](#)

Go语言项目架构：

[Go项目架构指南 - 云+社区 - 腾讯云 \(tencent.com\)](#)

[《阿里巴巴Java开发手册v1.4.0（详尽版）》更新，新增16条设计规约-阿里云开发者社区 \(aliyun.com\)](#)

Http跨域访问参考：

[POST请求变成OPTIONS的原理及处理 - 掘金 \(juejin.cn\)](#)

[OPTIONS - HTTP | MDN \(mozilla.org\)](#)

[\(34条消息\) golang跨域访问_benben的博客-CSDN博客_go 跨域](#)

Dijkstra算法参考：

[Dijkstra算法详解 通俗易懂 - 知乎 \(zhihu.com\)](#)

Ant Design参考：

[Ant Design - The world's second most popular React UI framework](#)

[介绍 — Vue.js \(vuejs.org\)](#)