

# 数据结构课程设计文档

组号	班级	姓名	学号
	2022211305	胡宇杭	2022212408
	2022211305	孟林	2022210484
	2022211305	陈炳璇	2022211479

## 软件开发任务的描述

游学管理系统，是一个综合性的服务学生游学的线上平台。旨在帮助同学进行目的地查询，路线规划，内部概览，日记管理等一系列功能。能够服务到出发前，游玩时，回程后三个时间段，是一个一站式的游学数字平台，能够让同学们做到游有所乐，学有所成，使同学们成长路上的好伙伴。

## 功能需求说明和分析

### 数据规模要求

1. 景区和校园数量至少 200 个（各 200 个），景区和校园内部可以重复。具体可以先做好一个作为测试用例，其余数据可以在此模板之上进行复制或修改，对于这些复制数据，在可视化展示时可以有数据图片不匹配的现象，即我们可以只使用测试用例的地图图片
2. 景区和校园内的建筑物（景点、教学楼、办公楼、宿舍楼）总数 不少于 20 个；其他设施不少于 10 种（商店、饭店、洗手间、图书馆、食堂、超市、咖啡馆等），总数量 不少于 50 个
3. 景区、校园内部的道路图
  - 建筑物、服务设施、道路转折点等需要作为节点存入图中
  - 边数不能少于 200 条
4. 系统用户数 不少于 10 人

### 基本功能需求

#### 游学推荐系统

1. 用户可以选择景点、学校作为目的地
2. 在游览前，系统会根据游学热度、评价、个人兴趣推荐游学景区和学校

3. 推荐过程中要求按照用户选择的指标（见上文）进行**不完全**排序，获取前 **10** 位景区或学校
4. 用户可以通过输入景区或学校的**名称、类别、关键字等**进行查询，系统应根据上文方法对查询结果进行排序

## 游学路线规划

1. 单点路径规划：进入景区或学校**后**，用户可以通过输入目标景点或场所信息来获取从当前位置出发到达景点或场所的最优游学路径
2. 多点路径规划：进入景区或学校**后**，用户可以通过输入多个目标景点或场所信息来获取从当前位置出发到达多个景点或场所的最优游学路径（**参观完后需返回当前位置**）

对于最优游学路径，有以下几种策略：

- 最短距离策略：保证所用距离最短
- 最短时间策略：假设每条道路拥有不同的拥挤度  $\delta$  ( $0 < \delta \leq 1$ )（可以随机生成）和理想速度，真实速度 = 拥挤度 \* 理想速度，保证经过的时间最短
- 交通工具的最短时间策略：见选做部分

## 场所查询

1. 在景区或学校**内部**时，选中某个景点或场所，可以显示周围一定范围内的**服务设施**，并根据距离排序
  - 可以通过类别对结果进行过滤
  - 可以通过输入类别名称查找服务设施

## 游学日记管理

1. 用户可以在游学**过程中或结束后**可以撰写游学日记，需要支持中文字符
2. 通过数据库对所有游学日记进行统一管理
3. 用户可以浏览和查询所有用户的游学日记，用户浏览完可以对游学日记进行评分
  - 可以按照日记热度（浏览量）、评价（评分）和个人兴趣进行推荐

## 游学日记管理

1. 用户可以输入游学目的地，对目的地的游学日记进行排序（排序规则同上）并显示
2. 用户可以输入游学日记的名称进行精确查找**要求在数据量较大、变化快的情况下进行高效查找**
3. 用户可以按日记内容进行全文检索
4. 可以对游学日记进行压缩存储（哈夫曼树）

## 选做功能需求（五选三）

1. 设计导航功能的图形界面，包括**地图展示和输出路径展示**

2. 室内导航策略：模拟教学楼的结构和景区内博物馆等建筑物的内部结构，进行室内导航，包括大门到电梯的导航、楼层间的电梯导航和楼层内到房间的导航
3. 交通工具的最短时间策略：校区内可以选择自行车和步行，选择自行车时，只能走自行车道路，默认自行车在校区任何地点都有；景区内可以选择步行和电瓶车，选择电瓶车时只能走电瓶车路线，电瓶车路线固定，默认上车即走；不同交通工具可以选择时，考虑不同拥挤度的情况下时间最短（时间最短的线路，可以是多种交通工具混合）
4. 采用推荐算法（基于内容推荐、协同过滤推荐）进行景点、学校、游学日记的推荐；
5. 使用 AIGC 算法根据拍摄的景点或者学校的照片进行游学动画生成

## 总体方案设计说明

### 软件开发环境

1. 开发环境及 IDE：
  - 前端：
    - 开发环境：Windows 11
    - IDE：VSCode
  - 后端：
    - 开发环境：MacOS
    - IDE：XCode
2. 编程语言及框架：
  - 编程语言：Vue3、C++
  - 框架：Boost
3. 构建工具：CMake
4. 版本控制：Git
5. 数据库管理：MySQL

### 总体结构

前端：前端部分负责与用户直接交互，通过提供直观且响应迅速的用户界面，让用户能够轻松地访问和使用系统的各项功能。前端提供的功能如下：

用户界面展示，交互逻辑设置，与后端进行数据沟通。后端部分则负责处理系统的业务逻辑，设计 API，存储数据以及算法实现

通过这样的划分，我们的游学管理系统能够在前端提供高质量的用户体验，同时在后端保证数据的准确性、业务逻辑的正确性和系统的稳定性。前端与后端的紧密合作，是我们提供优秀服务的基础。

后端：

后端部分由五个系统组成（除去网络系统），分别为：异常处理系统、消息处理系统、用户管理系统、游学系统、游学日记管理系统。具体每一系统对应处理的功能如下：

1. 异常处理系统：该系统会捕获其他系统中产生的所有异常并集中进行处理
2. 消息处理系统：对前端发送的消息进行解析并调用相应的方法进行处理
3. 用户管理系统：管理用户的注册、权限、登入、退出和销户
4. 游学系统：实现游学系统推荐、游学路径规划和场所查询功能
5. 游学日记管理系统：实现编写游学日记、日记压缩和存储、日记查询和文本查询功能

## 模块划分

### 前端

前端部分，我们的系统会展示首页和导航栏。其中首页的顶部会显示搜索栏，用户可以在此处搜索目的地。中部显示热门游学目的地推荐，而底部导航栏会分为两部分，一部分是目前所在的首页，也就是游学地点搜索和推荐的页面，另一页面则是游学日记管理页面。

首页在默认情况下会显示根据算法推荐的目的地，而点击进入搜索栏之后，用户则可以搜索目的地，键入回车则会展示按照排序算法排列的搜索结果。

而当用户通过推荐界面或者搜索界面选中目的地之后，则可以进入游学路线规划部分。根据用户当前的位置和用户选择的目的地，我们会展示最优路线，并加入策略选择。

在选中目的地之后，如果点击“详情”按钮，则会展示目的地内部的场所供用户查询，包括超市，公共卫生间等。用户也可以在景点内部选择游乐设施，系统会对选中的设施按照距离进行排序。用户还可以通过选择设施类别来对场所进行筛选。

最后是游学日记管理模块，可以通过底部导航栏进入。选中后会进入日记管理页面，默认展示推荐的游学日记。顶部导航栏依然是搜索，用户可以查找关键字，对日记库进行检索，页面则会展示按照推荐算法排序的日记。而在右下角还有一个画笔按钮，点击按钮则可以自己撰写日记，写完日记后可以点击上传。

### 后端

1. 消息处理系统：
  - 对前端发送的 Json 包进行解析
  - 根据解析出来的消息调用相应系统模块的方法处理
  - 保证线程安全
  - 将处理后的消息返回给前端
2. 游学系统：
  - 游学推荐模块：
    - 存储所有游学地点，同时保证对游学地点的高效查找
    - 根据地点热度、评价、用户个人兴趣对游学地点进行排序并显示前10位地点

- 用户可以通过类别筛选地点、并且支持对地点的模糊搜索功能
- 景点模块：
  - 存储该景点对应的设施建筑信息及对应的图
  - 游学路径规划模块：
    - 用户可以通过输入地点获得从当前位置到目标地点的最佳路径
    - 用户可以输入多个地点获得从当前位置出发游览所有地点并返回出发点的最佳路径
  - 场所查询模块：
    - 用户可以选中（或搜索）某个地点查找其附近一定范围的其他设施
    - 用户可以选择特定类型对设施进行过滤
    - 对查找到的地点根据距离进行排序

### 3. 日记管理系统：

- 日记编写模块：
  - 用户可以通过图形化界面撰写日记并保存
- 日记压缩存储模块：
  - 对游学日记进行无损压缩
  - 将压缩后的游学日记存储到数据库中
- 日记查询浏览模块：
  - 为用户推荐喜爱的日记
  - 用户可以输入日记名称进行精确查找
  - 用户可以输入地点查询关于该地点的游学日记
- 日记修改删除模块：
  - 用户可以浏览游学日记的内容并在文章中进行文本查找
  - 用户可以对游学日记的质量进行评分
  - 可以识别用户 ID 与游学日记的作者 ID 是否吻合，判断用户是否有权限修改删除日记
  - 同步删除用户的对应游学日记索引和数据库中对应的游学日记

## 各模块设计说明

### 1. 游学推荐模块：

- 存储所有游学地点：将游学地点存储进数据库中进行统一管理
- 根据地点热度、评价、用户个人兴趣对游学地点进行排序并显示前10位地点：构建大小为10的小根堆，通过重载运算符实现不同需求下的排序功能。同时，由于只是对前十个元素进行排序，算法的时间复杂度也远优于快速排序等方法，时间复杂度为  $\theta(n\log 10)$
- 通过类别筛选地点、并且支持对地点的模糊搜索功能：设想为使用 Trie 字典树或 倒排索引 或 BM 字符串匹配暴力求解，但 Trie 不支持模糊搜索功能，倒排索引的建立过于复杂，目前待定

### 2. 景点模块：

- 存储该景点对应的设施建筑信息及对应的图：设施建筑信息的存储采取直接存储到数据库中，并为每个设施建筑分配索引，在图中使用索引描述节点之间的关系；考虑到不需要频繁修改图，可以将图序列化后以文本（json）格式存入数据库中

- 合并游学路径规划模块和场所查询模块

### 3. 游学路径规划模块：

- 用户可以通过输入地点获得从当前位置到目标地点的最佳路径：由于我们构建的地图是具有物理意义的，所以我们可以存储每个节点的绝对坐标。由于两个节点间的欧式距离小于等于实际距离，因此可以当作启发式函数指导 A\* 寻路算法
- 用户可以输入多个地点获得从当前位置出发游览所有地点并返回出发点的最佳路径：这个问题可以分为两部分：
  - 计算出各个节点间的最短距离：最经典的方法是使用 Floyd 算法，但考虑到中途会引入许多无用计算（因为我们并不需要知道所有点之间的最短距离），因此，此处我们会同时测试使用 A\* 暴力求解所需点之间的最短距离和 Floyd 算法的实际消耗，并取最优方法
  - 解决旅行商问题：对不同数据规模采取不同算法进行求解
    - 较小数据规模时：采用状态压缩动态规划算法求解，时间复杂度为  $\theta(n^2 2^n)$
    - 中等数据规模时：使用分支限界法求解，通过适当地剪枝来压缩搜索空间，预期可以获得比动态规划更好的性能
    - 较大数据规模时：由于我们的系统需要实时将结果反馈给用户，因此这种时候再去寻求最优解并不是一个好的选择，因此采用贪心、模拟退火算法来需求近似解

### 4. 场所查询模块：

- 用户可以选中（或搜索）某个地点查找其附近一定范围的其他设施：通过 BFS 得到一定范围内的所有设施
- 用户可以选择特定类型对设施进行过滤：在搜索时对需求类型和实际设施类型比较并过滤不匹配项
- 对查找到的地点根据距离进行排序：由于 BFS 查询出的解在距离上有一定的顺序，所以我们选择综合评估希尔排序和快速排序的性能并选择最好的方案

### 5. 日记编写模块：

- 用户可以通过图形化界面撰写日记并保存：使用 `std::string` 存储日记，在撰写界面，考虑到用户可能在每一段之间来回切换修改，采用 `std::vector` 存储每一段的内容，以实现高效的撰写功能

### 6. 日记压缩存储模块：

- 对游学日记进行无损压缩：使用哈夫曼树对游学日记进行压缩
- 将压缩后的游学日记存储到数据库中：分别将压缩后二进制文件的日记、序列化后的哈夫曼树存入数据库

### 7. 日记查询模块

- 为用户推荐喜爱的日记：维护一个指向日记的指针数组，根据指定属性对指针数组进行快速排序
- 用户可以输入日记名称进行精确查找：使用日记名称构建一棵 Trie 树，实现对日记进行高效精确的查找
- 用户可以输入地点查询关于该地点的游学日记：遍历指针数组，找到对应地点的游学日记并显示

### 8. 日记浏览修改删除模块：

- 用户可以浏览游学日记的内容并在文章中进行文本查找：考虑到我们需要实现一个高并发的查询功能，可以创建一个“消息队列”，每隔一定时间就拿出其中的元素进行文本查找，这样，我们可以使用这些元素创建一个 AC 自动机，或者类比于 nextval 数组，创建一个 Trie 图，每次处理多个请求，达到加速处理的效果
- 用户可以对游学日记的质量进行评分：在浏览界面可以选择评分功能
- 可以识别用户 ID 与游学日记的作者 ID 是否吻合，判断用户是否有权限修改删除日记
- 同步删除用户的对应游学日记索引和数据库中对应的游学日记：Trie 树和数据库由于使用树状结构，删除较为容易，但指针数组只能通过遍历进行删除，因此这里使用一个 Hack：在指针数组排序时会先判断指向是否为 "nullptr"，如果是，就跳过该指针，直到用户下次进入程序才会真正刷新指针数组