

2025秋季学期-数据结构Project：面向自动驾驶导航的路径规划系统实现

负责助教：覃逸成、于海涛

2025秋季学期-数据结构Project：面向自动驾驶导航的路径规划系统实现

Background

Task

Description

Input

Output

Test Cases

Requirements

Hint

Extension

Deadlines

中期检查

最终提交

Assessment

- 中期检查 (20%)
- 代码核心功能实现 (50%)
- 开发文档 (30%)

Background

路径规划在实际应用中广泛用于导航系统、物流优化和游戏AI等领域，帮助高效计算从起点到终点的路线。然而看似复杂的实现实际上仍然绕不开最底层的数据结构，例如图论提供网络表示（如节点为城市、边为道路），而排序算法可优化路径搜索（如Dijkstra中使用优先队列）.....通过合理的数据结构与算法设计，路径规划系统能实现实时、动态的路径计算，支持权重调整和多目标优化。

Task

Description

在本项目中，你需要将本课程中所学知识进行合理运用，开发一个路径规划系统，支持加载地图文件，进行路径规划，并集成**不同时间段的模拟动态更新（如路况变化）**。

特别地，对于本次项目中地图道路的堵车情况实时计算，你需要使用一套**堵车系数计算公式**来处理CSV文件中的信息（可以参考现实案例与网上资料），例如简单的线性组合公式：

$$\text{堵车系数} = (\text{现有车辆数} / \text{车道数}) \times (\text{道路长度} / \text{道路限速})$$

Input

- 客户需求文件（TXT格式），记录了客户的出发地和目的地，每一次任务的出发地和目的地是不变的；
- 以时间命名的地图信息文件（CSV格式），一组测试用例的时间为同一天，文件中含有道路ID、起始地点名称、目标地点名称等多种信息（详见具体测试用例）。

Output

每当地图信息更新，输出一次，每次输出占一行，输出当前规划的最优路径（中间使用 --> 连接），例如：

```
复旦大学-->幸福小镇-->绿地足球场-->红星广场-->外滩-->海上购物城-->东方明珠-->陆家嘴
```

Test Cases

见eLearning任务发布的附件。

Requirements

- 使用C++/Java完成
- 使用任意你喜欢的工具开发：你可以使用任意编辑器/IDE，在任意平台（Windows/macOS/Linux）上进行开发。请在说明文档中注明你使用的开发环境/工具，并说明如何编译你的项目。
- 请注意：切勿抄袭！文档与代码提交后均会进行查重，一经检测到抄袭或雷同，落实后双方均作0分处理！

Hint

- 你可能需要处理一些**实际场景中可能遇到的特殊情况**以增强你的系统鲁棒性
- 注意内存消耗与时间开销
- 地图信息文件（CSV格式）中的数据并非要求全部用到
- **尽早动手！**课程PJ不同于Lab，工作量较大，建议不要在DDL前临时赶工，否则极有可能无法完成或是存在漏洞
- 建议在开始前**先思考中期检查中的问题**，并做项目整体的构思与设计

Extension

请注意：该拓展部分仅适用于学有余力的学生进行参考学习，与PJ评分无关！与PJ评分无关！与PJ评分无关！。

- 如果想要对堵车系数实现更精确估计，还可以：
 - 考虑该道路相邻道路的车辆数，以提前计算即将汇入该道路的车辆数，这会涉及到多元计算等更复杂的内容

- 引入机器学习预测路况（如未来车辆数），将预测值集成到权重计算中，当然这会引入潜在的数据结构需求（如时间序列存储）
 - 如果想要进一步提高系统运行的效率，还可以：
 - 使用例如 A* 算法（结合启发式函数，如欧氏距离）来加速搜索；
- A 搜索算法是一种在图形平面上，有多个节点的路径，求出最低通过成本的算法。常用于游戏中的NPC的移动计算，或网络游戏的BOT的移动计算上。该算法综合了Best-First Search和Dijkstra算法的优点：在进行启发式搜索提高算法效率的同时，可以保证找到一条最优路径。*
- 使用并行数据结构（如多线程优先队列）处理现实情况中的大规模地图（如城市群）
 - 加入缓存（Cache）机制，存储最近计算的最短路径，以避免重复计算
 -
 - 实际场景中的导航算法并非只输出一条最优路径，例如我们在现实生活中导航APP最终会提供多条路径供我们备选

Deadlines

中期检查

DDL: 2025年11月16日 23:59

提交pdf 格式的中期文档并将目前已完成的源代码打包成zip格式提交到eLearning。两个提交窗口分别提交，文件名格式要求请务必遵守：

- 文档：学号_姓名_中期文档.pdf
- 代码：学号_姓名_中期代码.zip

在中期检查时你应该已经搭建好项目的代码框架，未完成的代码可以用注释代替。

最终提交

DDL: 2025年12月7日 23:59

提交pdf 格式的开发文档并将可运行的最终版源代码打包成zip格式提交到eLearning。两个提交窗口分别提交，文件名格式要求请务必遵守：

- 文档：学号_姓名_开发文档.pdf
- 代码：学号_姓名_最终代码.zip

请确保你提交的最终版源代码能够通过编译并正常运行。

Assessment

1. 中期检查（20%）

在中期文档中需要思考并回答以下问题：

1. 总结所有你掌握的最短路径算法，无需对算法内容进行详述，只需要简要概括它们各自的核心思想、适用场景、时间复杂度、空间复杂度。

2. 在理想的静态路径规划中，Dijkstra算法是否总能保证最优路径？如果是，请**简要证明**其最优性；如果不是，请举例说明在哪些实际场景它可能需优化。
3. 比较稀疏图 vs 稠密图，分应用场景讨论一下哪种表示更高效，在本题场景中应该怎么选择，为什么？
4. 假设对一个地图文件进行了路径规划，如何准确还原/可视化原始路径？请说明Dijkstra算法回溯机制。
5. **讨论题**：本次课程设计中完成的这个系统，距离投入实际应用或者你预想中完美的路径规划系统，还缺乏什么，有什么思路完成（**不需要实际实现，可以参考Hint和Extension部分的提示**）？

此外，中期文档中还需要包括以下部分：

1. 对初步构思的项目架构进行介绍，分析可能遇到的困难或挑战（如果有）
2. 对目前的项目进度进行总结，并整理好后续规划
3. 附上目前完成的代码框架主体部分，并做简略的说明

若中期文档完成度较差，将视情况酌情扣分。

2. 代码核心功能实现（50%）

你的最终版代码应**至少实现以下功能**：

- **自行使用代码实现**一个较为完整的自定义Graph类，**基础接口齐全，扩展接口自行设计**以完成项目需求
- 你的路径规划系统能够**正确解析并处理输入的地图数据**
- 通过代码实现一套**合理的图权重（堵车系数）计算公式**，**可以参考题干给出的公式，也可以自行设计**
- 路径规划系统**能够通过你设计的算法进行计算并输出符合要求的每一轮规划的最优路径**
- **尽可能保证系统的鲁棒性**，能够处理遇到的特殊情况
- 在以上要求基础之上，尽可能提高代码运行效率

除此之外，**特别说明**：

- 对于你的算法中涉及排序内容的部分，你可以选择直接使用优先队列来处理，当然如果有能力建议自己实现排序算法。
- 对于自行设计的图权重（堵车系数）计算公式，**并非越复杂得分就会越高**

3. 开发文档（30%）

开发文档（PDF 格式）应**至少包含以下内容**：

- 代码结构概要说明
- 项目核心功能的设计、实现思路的介绍
- 开发环境/工具，以及如何编译/运行项目
- **任选一组测试用例，对照代码和调试截图，进行过程说明**
- 测试结果（表格记录每个测试用例的每一轮结果）
- 本次项目中遇到的问题和解决方案
- 其他你想说明的任何内容（若有）