

城市交通路径规划系统

一、简要信息

随着城市交通流量的不断增加，如何在确保交通流畅的同时，为多个车辆提供最优的出行路径成为一个重要的挑战。本项目旨在设计并实现一个**智慧城市交通优化系统**，系统通过对城市内不同地点之间的交通网络进行分析，**为多个车辆提供最短时间规划**，同时考虑**道路车流量限制**，以实现交通系统的整体优化。

★提交截止日期：2024.12.29 23:59

二、Project 要求

2.1 基本要求

(1) 实现基础数据结构

- 设计记录城市交通网络的数据结构
- 设计记录车辆信息的数据结构

(2) 为多个车辆提供最短时间规划

给定多个车辆的起始地点和目的地点，系统需求得所有车辆从起点到终点的时间之和最短。车辆每单位时间可以行进的道路长度为 1。

2.2 高级要求

(1) 考虑道路车流量限制

在进行路径规划时，需要考虑道路的最大车流量。如果某条道路的车流量超过了其最大承载能力(例如，某条道路已经饱和，无法再容纳更多车辆)，则该道路将被排除。因此，需要系统在基础要求的基础上增加对道路车流量的考虑，以确保交通网络的流畅性。

三、实现细节

3.1 输入格式

输入共包含路径规划的基本信息、地点之间的道路长度、地点之间的交通流量限制、车辆的信息这几个部分。上述几个部分依次输入，以下是详细说明：

(1) 路径规划的基本信息

输入包含一行共 2 个正整数，分别为地点数量 N ($1 \leq N \leq 20$) 和车辆数量 M ($1 \leq M \leq 20$)。

(2) 地点之间的道路长度

输入包含 N 行，每行包含 N 个用空格间隔开的数字。地点编号从 1 开始计算。输入的数字若为 0，则表示对应的两个地点之间没有道路连接；若为正整数，则表示对应的两个地点之间道路的长度。

(3) 地点之间的交通流量限制

输入包含 N 行，每行包含 N 个用空格间隔开的数字。地点编号从 1 开始计算。其中 N 为地点的数量。输入的数字均为正整数，表示同一时间，最多能同时行驶车辆的数量。对于基本要求，此处输入的交通流量限制为足够大的正整数（即不需要考虑交通流量限制）。

(4) 车辆的信息

输入包含 M 行，每行包含 2 个用空格间隔开的数字。M 行分别对应编号为 1 到 M 的车辆的信息。每行输入的第 1 个数字为车辆的出发地点编号，第 2 个数字为车辆的目的地编号。车辆每单位时间可以行进的道路长度为 1。

3.2 输出格式

输出共包含车辆的行驶路径和总的行驶时长两部分。

(1)车辆的行驶路径

输出 M 行，第 j 行对应编号为 j 的车辆的行驶路径信息。每一行包含车辆从其出发地开始到达目的地的完整路径。该路径使用 K 个数字表示，为该车辆经过的地点编号（包含出发地和目的地）。

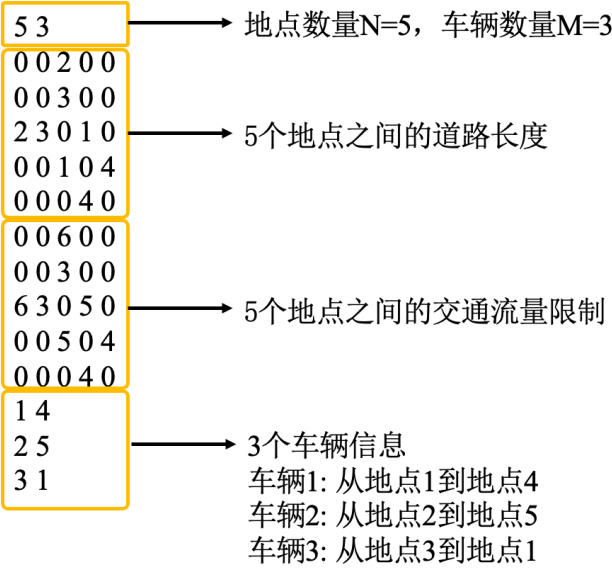
(2)总的行驶时长

输出 1 行，为所有车辆从出发地开始到目的地的总时长。

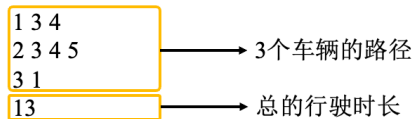
3.3 样例说明

测试样例

(1)输入



(2)输出



3个车辆的路径

车辆1: 车辆从地点1出发, 经过地点3, 到达地点4

车辆2: 车辆从地点2出发, 经过地点3和地点4, 到达地点5

车辆3: 车辆从地点3出发, 到达地点1

四、提交要求

作业通过 elearning 进行提交, 提交文件应为一个文件名为学号, 后缀名为 rar 或者 zip 的压缩包, 压缩包示例结构如下:

```
student-id.zip
├── report.pdf
├── code
│   ├── main.cpp
│   └── readme.txt (可选, 详见项目评分)
```

report.pdf 长度不宜超过 10 页, 正文字号行距等格式以清晰易读为佳, 内容包括解释整体实现思路、介绍已实现功能的思路、展示测试用例及结果、描述实现过程中收获(非必须)。若仅有报告, 无代码/抄袭代码/乱写代码, PJ 将酌情扣分。

在 code 文件夹中, 所有代码文件需要包含必要的中文注释, 所有代码文件请使用 UTF-8 编码提交。

在完成 PJ 过程中若参考了互联网资料或获得了同学的帮助, 请在报告的末尾注明, 如未注明但在评分时发现直接抄袭互联网资料或其他同学提交的代码等情况, PJ 将酌情扣分。

五、项目评分

5.1 评分组成

本项目的评分包括代码评分和报告评分。

5.2 代码评分

代码评分时会在 Ubuntu 22.04 环境下, 对源文件进行编译。请选择如下 3 种方式之一提交代码文件。

(1) 提交单个 C 语言源文件

对于该方式, 请将所有代码放在一个源文件中, 文件命名为 main.c。

编译时, 使用 gcc11 对 main.c 直接编译, 编译命令为 gcc -Wall -std=c17 -O2 main.c -o main。

(2) 提交单个 C++语言源文件

对于该方式, 请将所有代码放在一个源文件中, 文件命名为 main.cpp。

编译时, 使用 g++11 对 main.cpp 直接编译, 编译命令为 g++ -Wall -std=c++20 -O2 main.cpp -o main。

(3) 提交包含多个 C/C++语言源文件

对于该方式，请将所有代码文件放在 `code` 文件夹中，并在 `code` 文件夹中放置一个 `readme.txt` 文件作为编译命令的说明。`readme.txt` 文件内仅可包含用于编译的命令（每行一条编译命令），不能包含额外的文字说明，命令中只可以使用 `gcc`(版本为 `gcc11`)和 `g++`（版本为 `g++11`），其中用于编译的命令（不包含链接等命令）需使用 `-O2` 选项。编译时，会按照 `readme.txt` 中的命令依次执行。

➤ 如不能正常编译和运行程序，或程序不能正常读入相应的测试样例，均会有较大程度的扣分。对于每一个测试样例，程序运行时长不能超过 20 秒。

5.3 文档评分

在代码能正确编译运行，并按要求正确的前提下，报告能说明整个程序的设计思路即可获得全部分数，请不要在报告中加入冗长的说明或复制大段的代码，报告格式尽量规范。

5.4 其他说明

不限制使用 C++标准库，但禁止使用任何第三方库，使用第三方库的定义包括直接引入第三方库的源代码、调用编译好的二进制文件等。