

# 《程序设计与算法综合实习 I》 课程设计报告

题    目：	快递员出行问题
班    级：	软件工程 6 班
学生学号：	201925220603
学生姓名：	陈琳
指导老师：	梁 云
提交时间：	2020 年 4 月 17 日
成    绩：	

华南农业大学 数学与信息学院

# 一、需求分析

## 1、系统概述：

近年来，快递物流行业的迅速发展，使人们的生活发生了质的飞跃。然而，顺带而来的，是快递物流的效率问题。快递员出行是现代物流的核心环节之一，快递员出行是快递物流长途中临门的最后一步。快递员出行路线的规划关系着用户对物流服务的直接体验。对此，快递员出行的路线规划问题显得尤为重要。如何用尽可能短的路程走完所有的快递站点？如何选择交通堵塞情况较小的路线进行派件？如何选择环境污染程度较小的路线出行？如何尽可能降低快递员出行的能量消耗？如何在用户要求尽快送达货物的情况下，规划快递员所在地到用户所在快递站点的最短距离？如何实时更新交通情况有利于快递员的运行效率的提升？通过本系统的路线规划，能够满足以上所有的用户需求，大幅度提高快递员的出行效率，满足用户的及时收到快递货物的需求。

系统主要功能分两类：一个功能是快递站点数据，快递站点之间路径的数据的输入。另一个功能是最短路线的查询。快递员通过这个功能查询路程最短路线，污染最小路线，环境污染指数最小路线，交通堵塞情况最小路线，快递员所在点与目标点距离最短的路线，未派送的快递站点的路线规划问题。

## 2、系统运行环境：

### （1）实验硬件环境：

CPU :Intel(R) Core(TM) i7-8565U [CPU@1.80GHz](#) 1.99GHz

内存：8GB

显卡：集成显卡

### （2）实验软件环境：

操作系统：Window 10

DOS 系统支持中文输出

### （3）建议硬件环境：

CPU：奔腾 3 以上

内存：1G 以上

显卡：集成显卡

硬盘：10M

### （4）建议软件环境：

操作系统：windows 7/8/10

DOS 系统支持中文输出

## 3、功能需求描述：

(1) 用户数据导入模块：本模块是用户最新进入的模块，是进行路线规划的基础。数据导入分为两种模式，一种是使用默认文件的导入，在文件中默认含有“实验数据.txt”文件，内含设计好的实验数据，用户可以在里面更改数据；另一种是用户自己设计数据，通过程序弹出的输入提示，用户可以自行设计输入数据。

(2) 规划最短路径模块：本模块是此程序的核心模块。通过此模块，用户可以规划路程最短路线，消耗能量最小路线，环境污染指数最小模块，交通拥堵情况最小模块。用户必须在导入数据后才能进行路线的规划。

(3) 包含派件数为 0 的点的路线规划模块：用户可能会有派件数为 0 的快递站点的处理情况，本模块用于处理存在派件数为 0 的快递站点的情况。用于处理，快递员经过派件数为 0 的快递站点的路线路程或其他检测标准比未经过派件数为 0 的点更优的情况。

(4) 数据更新模块：本模块用于便利用户实时更新交通拥堵情况，做到高效地规划最有效的路线。

(5) 规划起始点到目的点两点之间的最短路线模块：本模块用于解决当收货人要求尽快收到货物是，规划快递员从当前所在站点规划到收货人的快递站点之间的最短路线。

(6) 规划未经过的点的遍历最短路线模块：当更新交通拥堵情况时，用户可以输入未派件的快递站点进行所剩快递站点的路线规划。

## 二、总体设计

### 1、总体开发思想：

#### (1) 主要的数据结构：

系统对数据操作的主要数据结构是图，数组，结构体。结构体中的每个数据用来储存与图有关的数据。其中，结构体内的数组 **BestWay** 中每个储存空间用于储存第几个经过的快递站点。数组 **Array** 第 *i* 个地址（此时 *i* 表地址序号）用于储存第 *i* 个快递站点（此时 *i* 表快递站点序号）的派件数目。数组 **flag** 第 *i* 个地址（此时 *i* 表地址序号）用于标记第 *i* 个快递站点（此时 *i* 表快递站点序号）是否已被派件。二维数组，此处以路程权重为例，数组的两个角标表示路程的两个端点的快递站点名，不存在的路径权重设为 $\infty$ ，代码中以“999999”表示 $\infty$ 。（如下图表格所示）

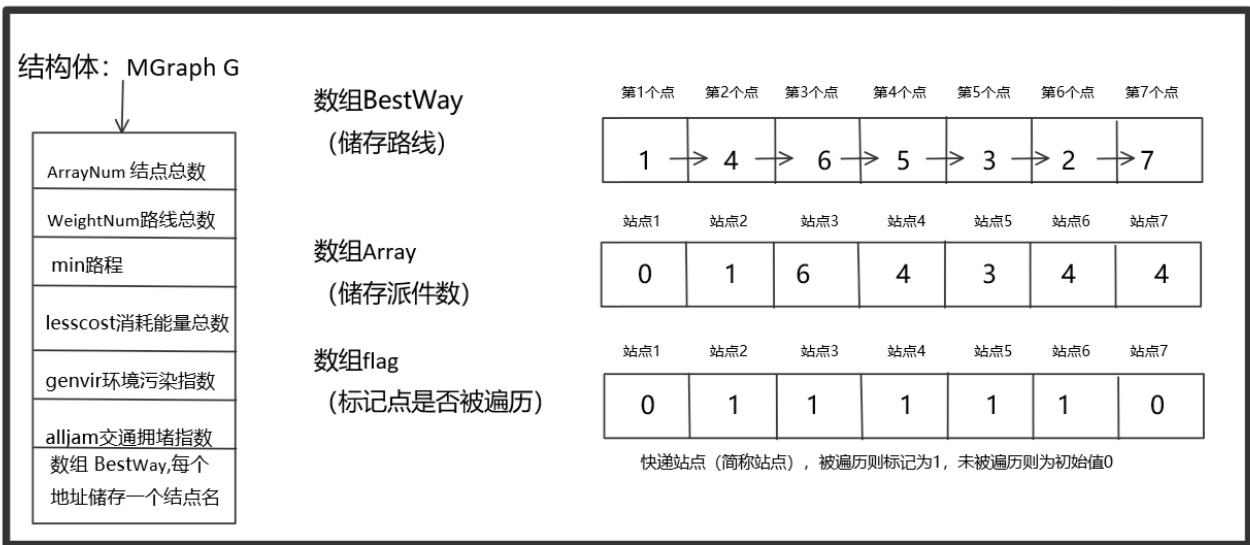


图 1 图的储存数据结构

图 1 中结构体包括快递站点总数（即节点总数），最优路线对应的路程，消耗能量总数，环境污染指数，交通拥堵指数，储存路线中快递站点遍历顺序的数组 **BestWay**。对于数组 **BestWay**，每确定完最优路线，第 *i* 个储存地址储存第 *i* 个经过的快递站点（例：**BestWay[1]**储存第 2 个经过的快递站点）。

二维数组，用于储存各权重（以路程举例）					
快递站点名					
快递 站 点 名	1	2	3	4	5
1	$\infty$	3	2	2	1
2	3	$\infty$	4	3	1
3	2	4	$\infty$	4	2
4	2	3	4	$\infty$	3
5	1	1	2	3	$\infty$

图 2 各权重的储存形式

权重都储存在类似图 2 的二维数组中。其中，图 2 表格内第 1 横行，第 1 纵行均表示快递站点名。其余小格通过对应它的快递站点名（即对应行，列首端的快递站点名）得出该权重的对应二维数组的 2 个角标。

## （2）数据存贮方式：

一维数组用于储存节点（即快递站点）的信息，包括各节点的派件数，该点是否被遍历的标记值，以及最优路线的顺序遍历结点储存，二维数组用于储存权重，包括快递站点之间路径的距离，消耗能量数，环境污染指数，交通拥堵指数。其中一维数组的角标表示该快递站点的序号名。二维数组的两个角标为路径两端的快递站点的序号名。

结构体用于储存该图的结点总数，路径总数，最优解的路程，环境污染指数，交通拥堵指数，能量消耗值。

## （3）使用的语言： C/C++

## 2、系统模块结构图：

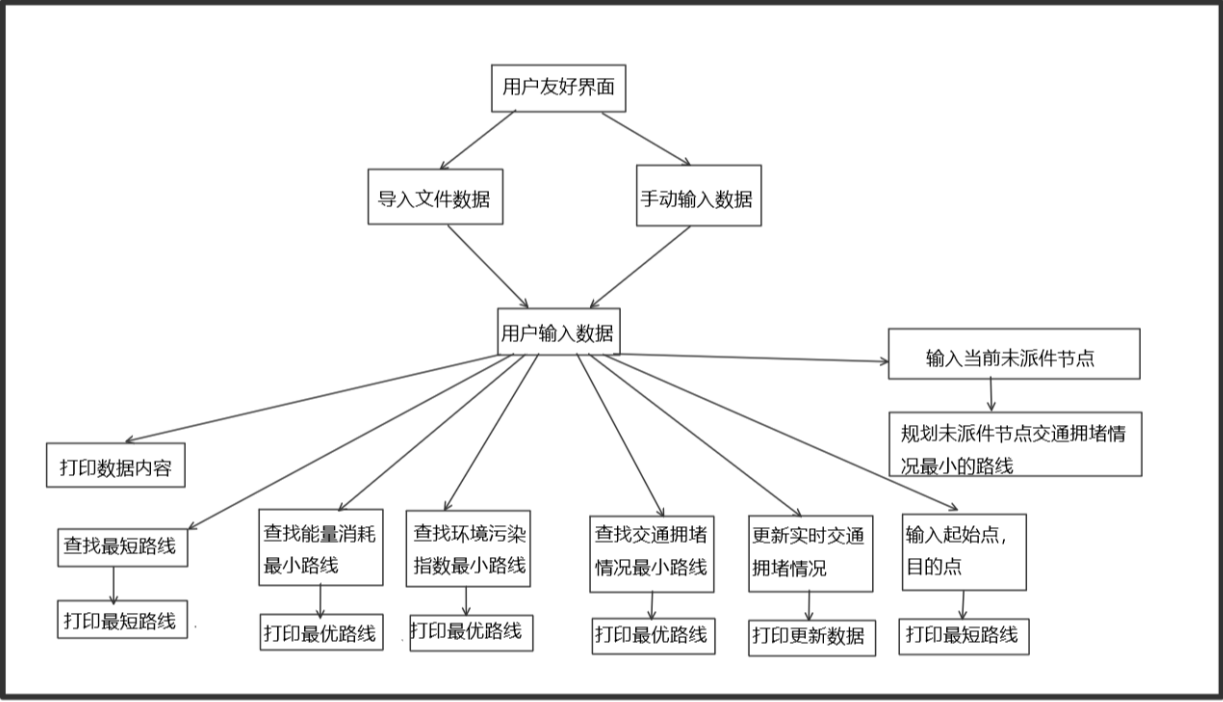


图 3 代码的功能执行步骤示意图

图 3 解析：代码运行后，马上出现用户友好界面，此时用户应当选择是导入文件内的数据还是用户手动输入数据，当数据输入完成后，用户选择模块功能任务，有打印数据内容任务（打印数据的详细内容）；查找各种最优路线并打印的功能；更新实时交通拥堵情况功能；规划两点之间最短路径的功能；规划未派件节点的功能。

3、模块设计： 以下为各个功能模块的流程图

(1) 规划路线模块

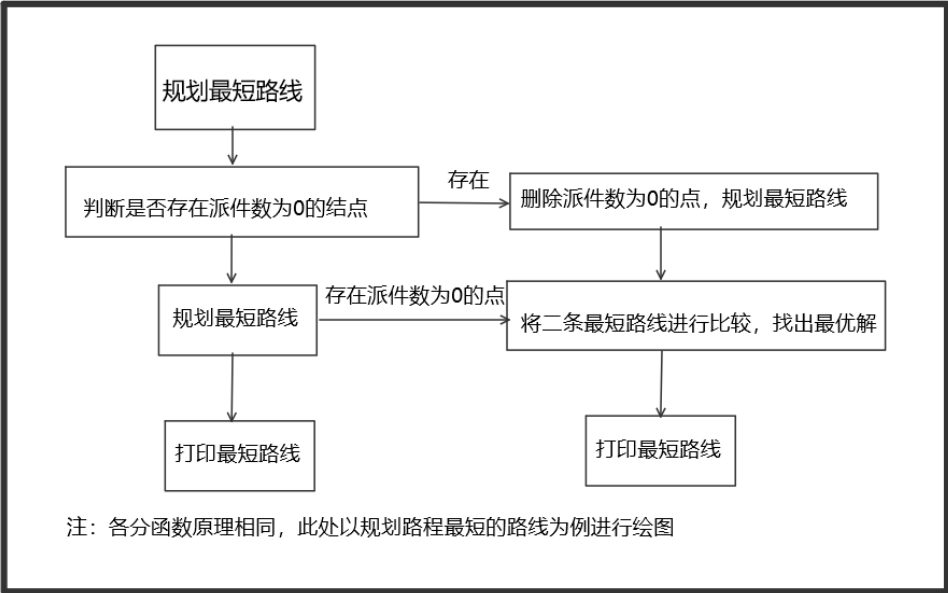


图 4 规划最优路径的图解

图 4 以规划最短路线为例，进行了最优路径的图解。当用户选择规划最短路径这一模块，系统自动判断用户输入的数据中是否存在派件数为 0 的快递站点。若不存在，则进行最短路线的规划，若存在，则在规划原有最短路线的基础上，删除派件数为 0 的点再构建一个图，进行最短路

线的规划，在综合所得的两个解，对比求得最优解。

(2) 储存路线模块

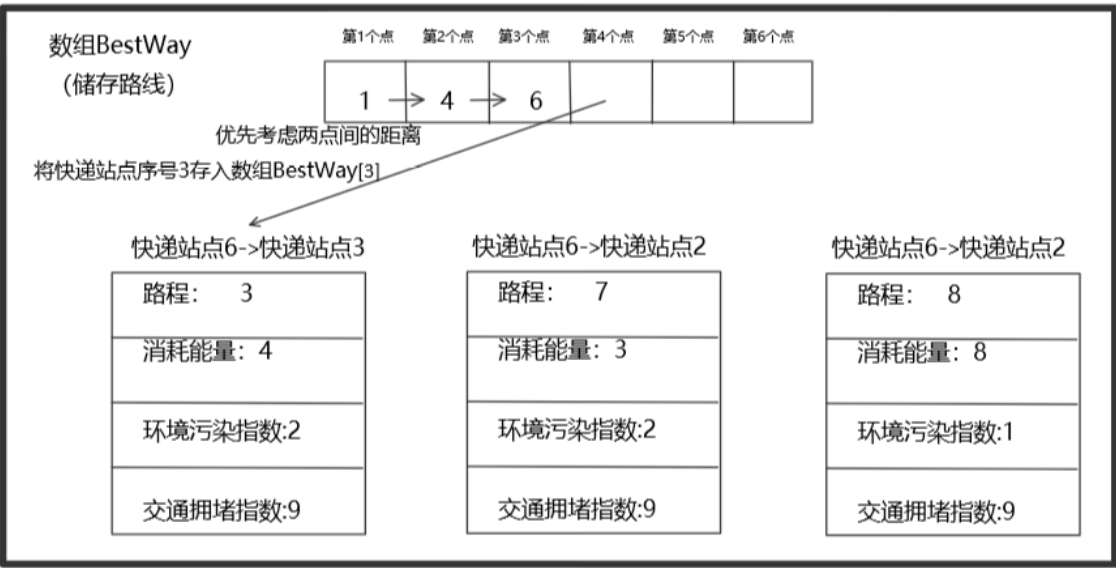


图 5 最短路线规划的步骤 1

最优路线的规划模块，分别规划以每个快递站点为起始点的最优路线，此处以快递站点 1 为起始点举例说明。每次系统保存上一个快递站点，调用与上个快递站点有关的所有权重值，如图 5 的快递站点 6 到快递站点 3 的各权重值。

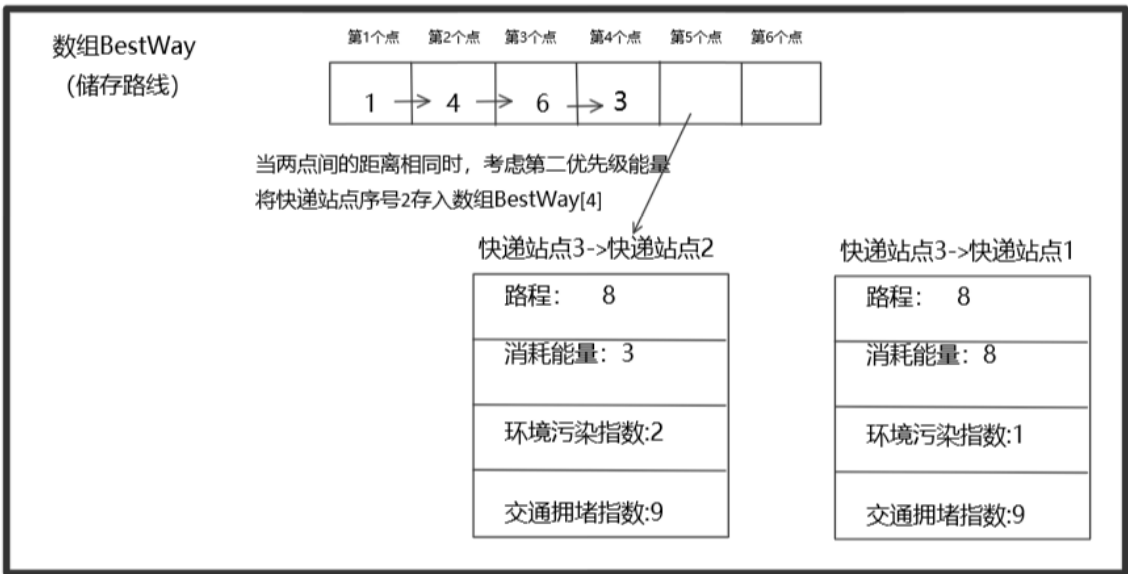


图 6 最短路线的规划步骤 2

最优路线的规划模块，按照优先级路程，消耗能量总数，环境污染指数进行选择。当两个结点对应的路程相同时，看下一个优先级消耗能量总数，当第二个优先级数量相同时，选择第三个优先级，环境污染指数，以此类推。当最优解规划完后，把当前最优解的路程与初始值路程进行对比，若当前最优解路程小于初始值路程，则令初始值为当前最优解，进行下一个结点的最优路线规划和与初始值的数据对比。

(3) 求全源最短路径模块

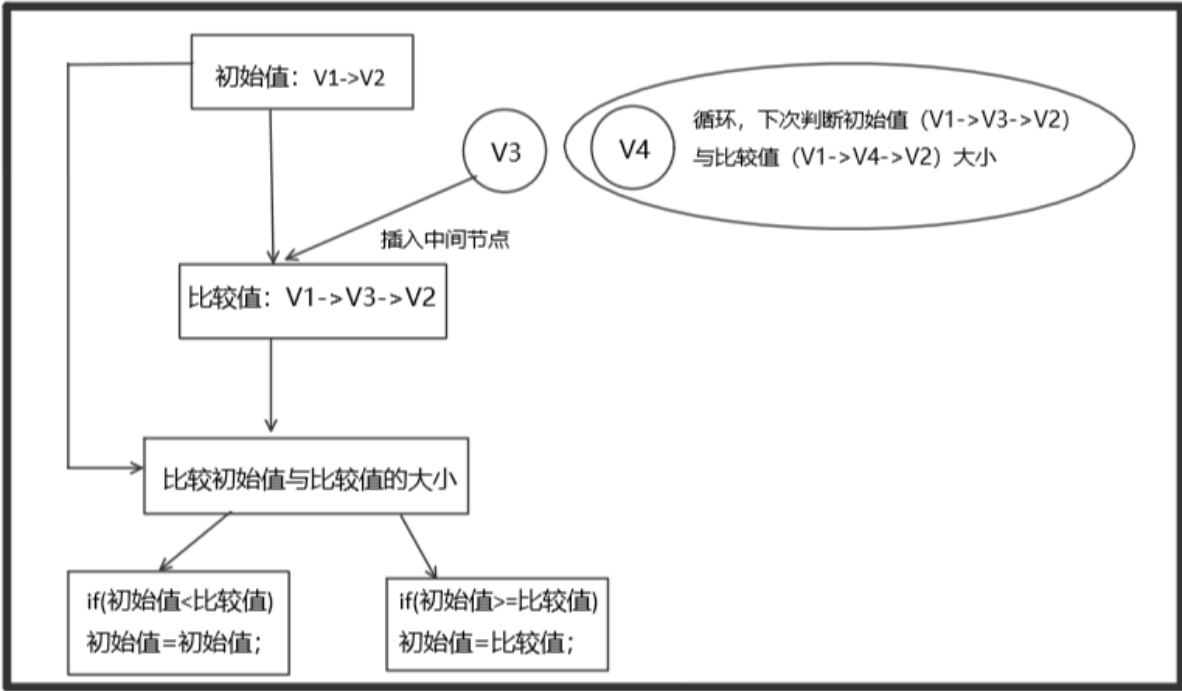


图 7 “全源最短” 路径规划 1

本模块思路与上一模块最短路线规划思维类似，都是通过将当前解与初始值进行对比，将最优解赋为初始值，通过循环进行下一次规划，比对。本模块通过在两个结点中不断插入中间结点，求得最优解。

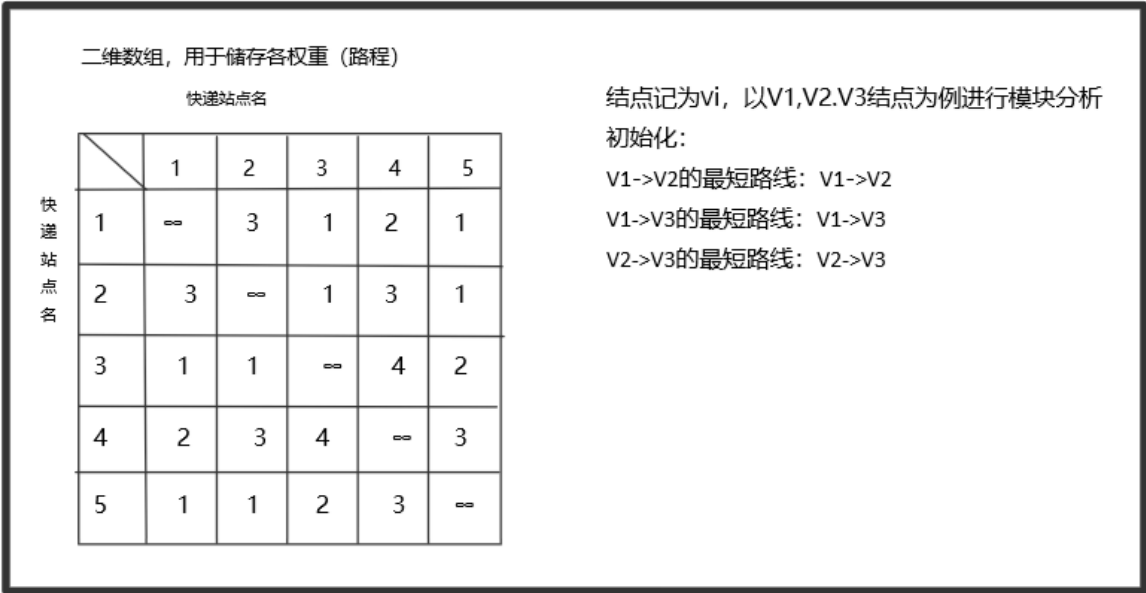


图 8 “全源最短” 路径规划 2

图 8 为函数赋予各节点的初始值以及路程权重的表示图。初始值两点之间的最短距离为两点之间的路径路程。若两点间不存在路径，则最短距离为路径无穷大 999999 值。





图 9 “全源最短”路径规划 3

在最短路径的比较中如果插入结点使得两结点之间的路程减小，则系统储存中间结点的名称，如图所示： $v1$  到  $v2$  若以  $v3$  为中间结点，会减少路程。系统储存  $v3$ 。同时，若  $v1$  到  $v3$  间的最短路线也有中间结点，则  $v1$  和  $v3$  的路线中储存中间结点 4。原来  $v1$  到  $v2$  的中间结点仍然只有  $v3$ ，不需加以改变。

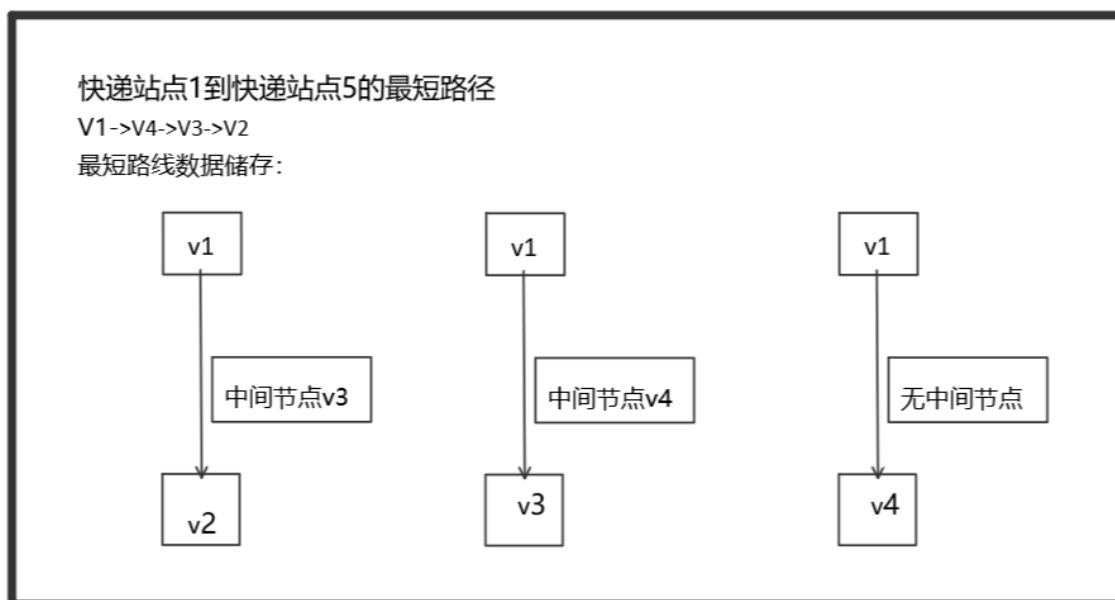


图 10 “全源最短”路径规划 4

图 10 表示在打印的过程中会使用递归函数：调用  $v1$  到  $v2$  之间的最短路线，发现存在中间结点，调用  $v1$  到  $v3$  之间的最短路线，发现  $v1$  和  $v3$  之间还存在中间结点  $v4$ ，调用  $v1$  到  $v4$  之间的最短路线，不存在中间结点，则从最底层的递归函数开始打印，打印出的结果即为  $v1 \rightarrow v4 \rightarrow v3 \rightarrow v2$ 。

## 三、功能实现

### 1、主要函数：函数名及参数含义；

#### (1) 界面主要函数

`void color (short x)` //颜色函数，x 为颜色序号，通过选择颜色序号，选择打印出来的字符颜色。

`void DrawPepper()` //打印欢迎界面，退出界面，界面是一直名为佩琪的小猪卡通形象，用于营造用户的友好界面。

`void friendly()` //打印主菜单界面，每次执行完任务都会弹出此界面，用户通过选择相应功能的序号，实现该功能的任务。

`void printInformation (MGraph G)` //打印用户导入的数据，即打印用户输入数据构建成的图的信息。G 表示用户输入信息构建成的图。

`void printPath(MGraph G)` //打印最短路径，当不存在派件数为 0 的快递站点或者存在派件数为 0 的快递站点，但是最优路径不经过该点。G 表示用户输入信息构建成的图。

`void printPath1(MGraph G,MGraph G1)` //打印最短路径，当距离最短的路线经过派件数为 0 的快递站点时。G 表示用户输入信息构建成的图，G1 表示用户输入信息删除派件数为 0 的点构成的图。

`void printCost(MGraph G)` //打印消耗能量最少路径，当不存在派件数为 0 的快递站点或者存在派件数为 0 的快递站点，但是最优路径不经过该点。G 表示用户输入信息构建成的图。

`void printCost1(MGraph G,MGraph G1)` //打印能量消耗最小路径，当能量消耗最小的路线经过派件数为 0 的快递站点时。G 表示用户输入信息构建成的图，G1 表示用户输入信息删除派件数为 0 的点构成的图。

`void printEnvir(MGraph G)` //打印环境污染指数最小路径，当不存在派件数为 0 的快递站点或者存在派件数为 0 的快递站点，但是最优路径不经过该点。G 表示用户输入信息构建成的图。

`void printEnvir1(MGraph G,MGraph G1)` //打印环境污染指数最小路径，当环境污染指数最小的路线经过派件数为 0 的快递站点时。G 表示用户输入信息构建成的图，G1 表示用户输入信息删除派件数为 0 的点构成的图。

`void printJam(MGraph G)` // 打印交通拥堵指数最小路径，当不存在派件数为 0 的快递站点或者存在派件数为 0 的快递站点，但是最优路径不经过该点。G 表示用户输入信息构建成的图。

`void printJam1(MGraph G,MGraph G1)` //打印交通拥堵情况最小路径，当交通拥堵情况最小的路线经过派件数为 0 的快递站点时。G 表示用户输入信息构建成的图，G1 表示用户输入信息删除派件数为 0 的点构成的图。

`void PrintFloyd(int i,int j)` //打印快递员所在的起始点到收货人所在点之间的最短路线。

#### (2) 功能实现函数：

`typedef struct ArrayName` //建立结构体，便于图的构建。

`int Judge(MGraph &G)` //判断所有节点中是否存在派件数为 0 的结点，有则返回 1，无则返回 0。

`void CreatGraph(MGraph &G)` //按照用户的文件输入数据构建的图以及数组。G 表示用户输入文件信息构建成的图。

`void CreatGraph1(MGraph &G1)` //将用户文件输入数据删除派件数为 0 的构建的图。构建而成的图 G1。

`void BuildTree(MGraph &G)` //按照用户的手动输入数据构建的图以及数组。G 表示用户输入信息构建成的图。

`void BuildTree1(MGraph G,MGraph &G1)` //按照用户的手动输入数据删除派件数为 0 的点构建的图以及数组。G 表示用户输入信息（未删除派件数为 0 的点）构建成的图。G1 表示用户输入信息（删除派件数为 0 的点）构建成的图。

`int Hanmilton(MGraph &G)` //根据用户输入的数据进行最短路径的查找。G 表示用户输入信息构建成的图。查找完返回最短路径。

`int Hanmilton1(MGraph &G1)` //根据用户输入的数据(删除派件数为 0 的快递站点)进行最短路径的查找。G1 表示用户输入信息（删除派件数为 0 的快递站点）构建成的图。查找完返回最短路径。

`void GoodCost(MGraph &G)` //根据用户输入的数据进行消耗能量最少路线的查找。G 表示用户输入信息构建成的图。

`void GoodCost1(MGraph &G1)` //根据用户输入的数据(删除派件数为 0 的快递站点)进行消耗能量最少路线的查找。G1 表示用户输入信息（删除派件数为 0 的快递站点）构建成的图。

`void GoodEnvironment(MGraph &G)` //根据用户输入的数据进行环境污染指数最小路线的查找。G 表示用户输入信息构建成的图。

`void GoodEnvironment1(MGraph &G1)` //根据用户输入的数据(删除派件数为 0 的快递站点)进行环境污染指数最小路线的查找。G1 表示用户输入信息（删除派件数为 0 的快递站点）构建成的图。

`int LessJam(MGraph &G)` //根据用户输入的数据进行交通拥堵指数最小路线的查找。G 表示用户输入信息构建成的图。返回交通拥堵指数。G 表示用户输入信息构建成的图。

`int LessJam1(MGraph G,MGraph &G1)` //根据用户输入的数据(删除派件数为 0 的快递站点)进行环境交通拥堵指数最小路线的查找。返回交通拥堵指数。G 表示用户输入信息构建成的图。G1 表示用户输入信息（删除派件数为 0 的快递站点）构建成的图。

`void updata(MGraph &G)` //用于更新用户新输入的实时交通拥堵情况。G 表示用户输入信息构建成的图。

`void Floyd(MGraph G)` //用 Floyd 算法计算全源最短路径，用于实现附加功能，快递员起始点到收货人终点的最短路线。

`void remain(MGraph &G,int alnum)` //规划当前未遍历快递站点交通拥堵情况最小的路线。G 表示用户输入信息构建成的图。alnum 表示未遍历的快递站点总数。

## 2、使用界面

(1) 欢迎/退出画面：

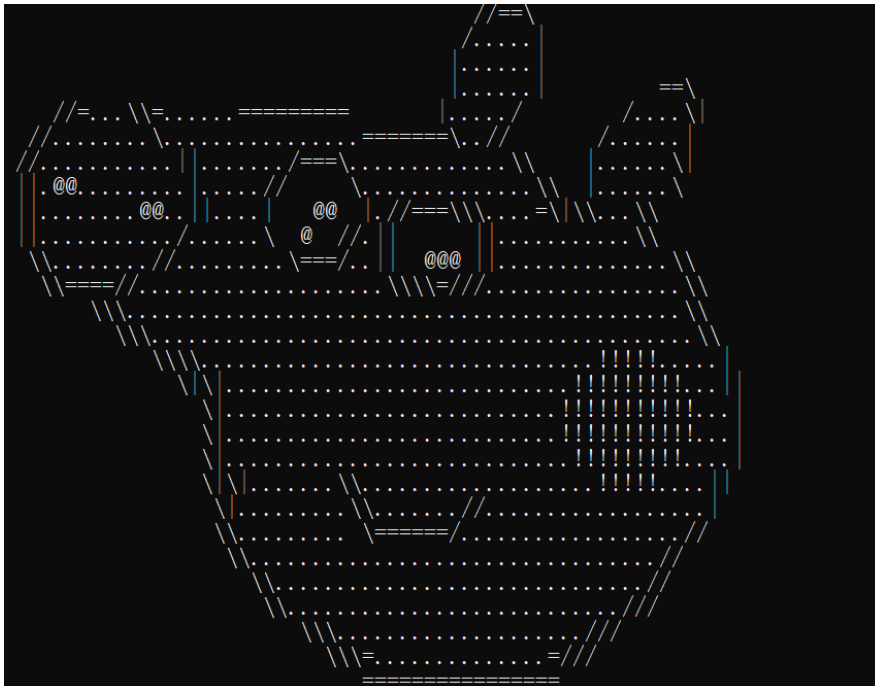


图 11 欢迎界面

运行代码，马上打印出欢迎界面，是一只名为佩琪的卡通小猪图像。接着弹出数据输入窗口，在此期间，用户不需要执行任何操作，系统会自动弹出。

## (2) 数据输入界面:

```
-X-X-X-X-X-X-X-X-X-X-X-X-X-X- Hello ,欢迎来到小猪快递路线设计程序 -X-X-X-X-X-X-X-X-X-X-X-X-X-X-
      用户须知: 您输入的数据必须保证 (删除派件数为0的快件站点之后),
                  以任何一点作为出发点都能把剩下的结点构成回路;
      [1] 输入001, 用户使用储存在"C:\课设终极版本\实验数据.txt"中的数据;
      [2] 输入002, 用户输入新的数据进行线路规划;
-X-X-X-X-X-X-X-X-X-X-X-X-X-X-
请输入:
```

图 12 数据输入界面

输入数据界面, 如果用户需要导入文件中的数据, 按 001, 回车键, 即可导入数据并弹出主菜单。如果用户需要手动输入数据, 则请按 2, 后续会有相应的指引输入操作。

```
-X-X-X-X-X-X-X-X-X-X-X-X-X-X- Hello ,欢迎来到小猪快递路线设计程序 -X-X-X-X-X-X-X-X-X-X-X-X-X-X-
      用户须知: 您输入的数据必须保证 (删除派件数为0的快件站点之后),
                  以任何一点作为出发点都能把剩下的结点构成回路;
      [1] 输入001, 用户使用储存在"C:\课设终极版本\实验数据.txt"中的数据;
      [2] 输入002, 用户输入新的数据进行线路规划;
-X-X-X-X-X-X-X-X-X-X-X-X-X-X-
请输入:2
骑手您好, 请您输入派件总数和派件可经过的路径总数:
(例: 20 20)
3
3
请您输入第1个快递站点的派件数目,
(例: 3)
1
请您输入第2个快递站点的派件数目,
(例: 3)
5
请您输入第3个快递站点的派件数目,
(例: 3)
0
请您输入第1个快递站点和第2个快递站点的路径数据, 请按照顺序为"快递站点 快递站点 路程 消耗能量 污染指数 交通拥堵情况"输入,
(例: 1 2 20 3 2 4)
1 2 1 3 4 5
请您输入第2个快递站点和第3个快递站点的路径数据, 请按照顺序为"快递站点 快递站点 路程 消耗能量 污染指数 交通拥堵情况"输入,
(例: 1 2 20 3 2 4)
2 3 2 5 6 4
请您输入第3个快递站点和第4个快递站点的路径数据, 请按照顺序为"快递站点 快递站点 路程 消耗能量 污染指数 交通拥堵情况"输入,
(例: 1 2 20 3 2 4)
1 3 2 4 6 4
-X-X-X-X-X-X-X-X-X-X-X-X-X-X- 骑手您好, 您的文件数据已为您导入完毕 -X-X-X-X-X-X-X-X-X-X-X-X-X-X-
```

图 13 用户手动输入数据界面

本系统可以完成 20 个以上的快递站点, 30 个以上的数据信息处理, 本处为了方便用户浏览报告, 使用 3 个快递站点。如有不便之处, 请见谅。

## (3) 功能界面: (注: 在执行完分功能, 总菜单界面都会弹出)

```
-X-X-X-X-X-X-X-X-X-X-X-X-X-X- 骑手您好, 您的文件数据已为您导入完毕 -X-X-X-X-X-X-X-X-X-X-X-X-X-X-
      用户须知: 您输入的数据必须保证 (删除派件数为0的快件站点之后),
                  以任何一点作为出发点都能把剩下的结点构成回路;
      [1] 输入001, 打印保存文件中所有数据构成的位置关系;
      [2] 输入002, 规划派件路程最短的路线详情;
      [3] 输入003, 规划派件消耗能量最小的路线详情;
      [4] 输入004, 规划派件污染指数最小的路线详情;
      [5] 输入005, 规划派件中交通拥堵情况最小的路线详情;
      [6] 输入006, 更新实时交通拥堵情况;
      [7] 输入007, 规划剩下未经过的快递站点交通拥堵最小情况;
      [8] 输入008, 规划当前位置到某一快递站点最短距离的路线;
      [9] 输入000, 退出当前程序;
-X-X-X-X-X-X-X-X-X-X-X-X-X-X-
```

图 14 用户界面主菜单

用户可以通过选择功能对应的数字, 输入该数字并按下回车键, 则该功能会运行并打印。













[illegible]

图 23 文件处理

用户可以通过选择是否清空文件内容来对文件进行处理。

## 四、总结

在 3 月初，我的实验课题分配到的是快递员出行规划。开始时，我并不知道从何下手，毕竟我从未学过规划路线的内容。而且这次课程设计与上次的折半查找的功能很不一样，我发现链表对于解决这个问题还不如数组来得直接。要实现这个程序，需要我进行深入的思考。

后来，我尝试着翻阅后面即将学到的数据结构中图的有关知识，同时，通过与同学的初步讨论，了解到“旅行商人问题”的路线规划问题和本程序很相似，我查阅算法中“旅行商人”问题的解决方案。经过一个星期的讨论，我初步将实现的功能进行分割，逐步精化、细化。考虑各种题目要求，以及用户输入数据时会使程序陷入死循环或者得出结果不是最优解的可能。直到将题目确认分析完毕后，我们才各自选择适合自己的数据结构，进行编写程序。在编写程序的过程中，我们遇到之前没有考虑周全的问题，再进行实时的讨论，不断优化用户体验。代码提交上去，得到老师的及时反馈，我再根据自己问题与别人相比稍微逊色的地方进行修改优化。

再谈谈具体遇到的问题。首先是选择什么数据结构储存数据的问题中。我先后考虑过数组，结构体，链表等数据结构。数组调用性强，很容易通过改动角标获得所要的值，但是返回数组较难以及数组内存受到函数限制（题目要求测试数据不少于 30 个，也就是  $30 \times 30$  的二维数组）；结构体可以储存各种类型不同的数据，可操作性强；链表构建过程繁杂，容易出错。最终我选择了一结构体和数组同时使用的方案。数组传输困难，以及受到函数的内存限制，我就改进代码，把数组定义在函数外面，这样即可以不用在声明函数写下数组，可以直接调用数组；同时，数组内存不受到函数限制，更好地满足了题目需求。此外，结构体方便了最优路线数组，最短路线值等的调用。如果没有结构体，在函数声明以及调用那里，就要写下这么一串数据，有了结构体，就可以用一个结构体代替这么一串数据，可谓是方便了代码的编写。

通过对储存数据的数据结构的选择，我收获了很多。以往 C 语言或者数据结构课堂的学习，我都是按照课本或者题目的要求对某一明确的数据结构进行编写代码。可是这次，数据结构是我自己根据编写代码需求选择适合自己的数据结构，这让我第一次认真分析各种结构的优劣，改变以往纯粹应试型的数据结构编写代码。在分析的基础上，对于以往的编写习惯有了优化，来改变这一结构原有的劣处，我想这个对于我来说是收获最大的。

比较困难的还是对派件数为 0 的点的考虑。因为如果按照原来思路对派件数为 0 的节点进行考虑，可能出现经过派件数为 0 的路线反而比未经过派件数目为 0 的路线更优；也会出现删除掉 0 结点后，会出现从派件数为 0 的点出发陷入死循环的可能性（因为查找我是从每个节点出发，找到从每个结点出发的最优解，再将所有最优解比较，得到真正这道题所求的解）。所以后期，我考虑删除派件数为 0 的点进行规划路线，同时考虑保留派件数为 0 的节点进行考虑，最后将二者进行比较，得出最优解。

总体来说，这次课程设计对我来说是一次能力的考验，同时也是一次收获满满的学习之旅。通过这次课设，我第一次站在用户的角度考虑友好界面，第一次考虑输入的数据需要给予的提示以及测试样例。同时也了解到了例如“`window.h`”的头文件，了解到了如何改变输出字体的颜色。同时，改变了以往我对图这一数据结构一知半解的状态，而是深入地了解这一数据结构并且能够根据自己的需求构建图，完善结构图的相关结构。对我而言，这次思维转变，知识的深入了解，都是难得可贵的。都给予了我莫大的收获！

评分表	
题目：快递员出行规划	
学号：201925220603	姓名：陈琳
评分项目	分数
1. 功能是否满足要求（50 分）	
2. 界面是否美观（10 分）	
3. 所学知识在设计中的运用是否得当（10 分）	
4. 对效率要求较高的部分是否有更深的思考与设计（10 分）	
5. 答辩情况（20 分）	
总成绩：	
评语：  系统代码实现：完成题目要求的主要功能，快递出行多种选择方案等均实现； 界面设计：简单、美观、直接； 知识运用：运用了链表、结构体、数组； 效率方面：使用排序，代码清晰明了，注释得当准确； 报告撰写：符合规范，功能设计合理，算法流程恰当，结果描述准确； 答辩情况：  其他：系统有一定复杂度和代码量，系统操作友好，提示准确丰富。 拓展多种路径搜索算法，设计了多种选择共存时，最优路径方案的选择，探索了派件点为 0 的节点出路。	
评阅老师：	评阅时间： 年 月 日