

数据结构

课程实验报告

|  |  |
| --- | --- |
| 题目： | 行车路线 |
| 学生姓名： | 刘大卫 |
| 学生学号： | 201826010215 |
| 专业班级： | 软件1802 |
| 完成时间： | 2019.12.10 |

1. **需求分析**
2. **问题描述**

小明和小芳出去乡村玩，小明负责开车，小芳来导航。

小芳将可能的道路分为大道和小道。大道比较好走，每走1公里小明会增加1的疲劳度。小道不好走，如果走小道，小明的疲劳值会快速增加，走s公里小明会增加s2的疲劳度。所有的小道不相交。

例如：有5个路口，1号路口到2号路口为小道，2号路口到3号路口为大道，3号路口到4号路口为大道，4号路口到5号路口为小道，相邻路口之间的距离都是2公里。如果小明从1号路口到5号路口，则总疲劳值为22+2+2+22=4+2+2+4=12。

现在小芳拿到了地图，请帮助她规划一个开车的路线，使得按这个路线开车小明的疲劳度最小。

1. **问题分析**

本题要求实现一个最短路径，其中采用的方法是遍历法，即每查询到疲劳值最小的元素，就将进行下一步。本题的逻辑上并不困难。

1. **输入数据**

输入的第一行包含两个整数n, m，分别表示路口的数量和道路的数量。路口由1至n编号，小明需要开车从1号路口到n号路口。

接下来m行描述道路，每行包含四个整数t, a, b, c，表示一条类型为t，连接a与b两个路口，长度为c公里的双向道路。其中t为0表示大道，t为1表示小道。保证1号路口和n号路口是连通的。

1. **输出数据**

输出一个整数，表示最优路线下小明的疲劳度。

1. **测试样例**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 2 | 3 |
| 输入 | 6 7 1 1 2 3 0 2 3 2 0 1 3 30 0 3 4 20 0 4 5 30 1 3 5 6 0 5 6 1 | 6 7 1 1 2 4 0 2 3 6 0 1 3 20 0 3 4 10 0 4 5 30 1 3 5 25 0 5 6 15 | 2 1 1 1 2 3 |
| 输出 | 48 | 75 | 9 |
| 设计目的 | 普通测试样例，只进行单次的查找。 | 更改权值。 | 查看只有两个点时是否正确。 |

1. **概要设计**
2. **抽象数据类型**

为实现上述程序的功能，我们使用了实验4设计的邻接表的ADT。使用各种函数进行查询转置等操作。抽象数据类型设计：

数据对象：n个节点，m条边；

数据关系：我们完成的是最短路径，所以其中的各种数据是线性关系。

基本操作：基本操作为线性表的操作：void init(int n)初始化邻接表；void getmap(int m)建表；int n()获取节点个数；int e()获取边数；int djk(int st, int en)获取从st到en的最短路径。

1. **算法的基本思想**

本题的主要目的是完成最短路径的设计，主要的设计并不复杂。

主要算法如下：

1. 输入节点数，边数；
2. 建表；
3. 采用djk算法完成输出。进行上述算法则可以成功解决问题。
4. **程序的流程**

本题是简单的输入、处理与输出模型。按照“一、需求分析”中的输入输出格式进行即可。

1. **详细设计**
2. **物理数据类型**

具体的物理数据类型我们选择的是实验1实现的链表，我们的实现基于课本上实现的ADT，下面给出具体的物理结构：

class Graph{

private :

// 拷贝函数

void operator = (const Graph &) {}

Graph(const Graph &) {}

public :

// 构造函数

Graph() {}

//析构函数

virtual ~Graph() {}

// 初始化函数

virtual void init(int) = 0;

// 返回节点数

virtual int n() = 0;

// 返回边数

virtual int e() = 0;

};在此基础上进行图的定义。

1. **输入输出的格式**

采用直接入。

数据格式如下：

输入的第一行包含两个整数n, m，分别表示路口的数量和道路的数量。

接下来m行描述道路，每行包含四个整数t, a, b, c，表示一条类型为t，连接a与b两个路口，长度为c公里的双向道路。其中t为0表示大道，t为1表示小道，以空格进行分隔。

1. **算法的具体步骤**
2. 声明一个图Graphm<int> \*mmp = new Graphm<int>;；
3. 输入数据；
4. 创建图；
5. 使用djk算法输出最小疲劳值；
6. **算法的时空分析**
7. 建立邻接表时顺序读入值，时间复杂度是；
8. djk算法；
9. **调试分析**
10. **调试方案设计**

调试目的：

测试程序是否可运行，发现代码的语法错误、连接错误、逻辑错误和运算错误，是否有不严谨之处。

测试样例：

6 7

1 1 2 3

0 2 3 2

0 1 3 30

0 3 4 20

0 4 5 30

1 3 5 6

0 5 6 1

调试计划：

设定好断点，单步执行，查看图中元素的变化情况，确定当前的执行环节，想办法对算法进行优化。对各种情况进行完善，确定边界情况。寻找问题，检查遗漏，及时进行修正，对代码的不完善之处进行修改。

1. **调试过程和结果以及分析**
2. **实验日志**

2019年12月10日：

成功解决。

通过本题，我对于合理应用数据结构，提升算法有了更加深入的理解，也勾起了我对于数据结构学习的热情，相信我在这个学期中，一定可以学好数据结构，并且把他应用到以后的实践中去。更好的提升自己。