6图的操作和应用

图结构的学习包括概念、定义、操作和编程。

通过“景区信息管理系统”的编程实践，学习图的遍历、Dijkstra算法、最小生成树、Prim算法和他们的编程应用。

开发和学习“景区信息管理系统”，达到如下目标：

(1) 掌握图的定义和图的存储结构

(2) 掌握图的创建方法

(3) 掌握图的两种遍历方法

(4) 理解迪杰斯特拉(Dijkstra)算法

(5) 理解最小生成树的概念和普里姆(Prim)算法

(6) 掌握文件操作

(7) 使用C++语言，利用图的数据结构，开发景区信息管理系统。

1、功能需求

现有一个景区，景区里面有若干个景点，景点之间满足以下条件：

条件一：某些景点之间铺设了道路（ 相邻 ）

条件二：这些道路都是可以双向行驶的（ 无向图 ）

条件三 ：从任意一个景点出发都可以游览整个景区（ 连通图 ）

开发景区信息管理系统，对景区的信息进行管理。 使用图的数据结构 来保存景区景点信息，为用户提供创建图、查询景点信息、旅游景点导航、搜索最短路径、铺设电路规划 等功能。

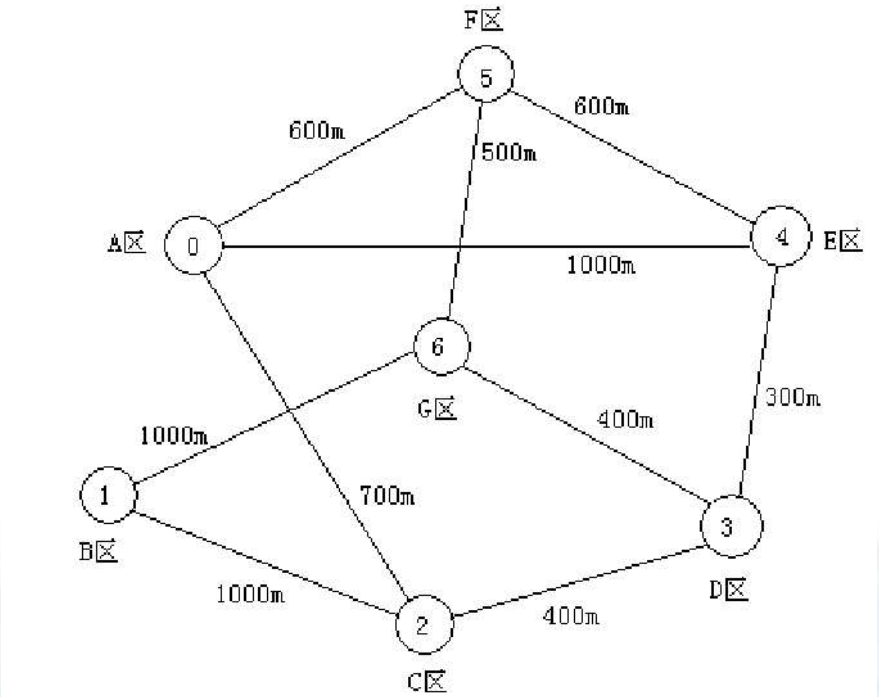


图6-1 景区和景点信息

（1）景点数据

景区的数据包含 景点信息 和景点之间的 道路信息 。分别由两个文本文件存储。Vex.txt文件用来存储景点信息；Edge.txt文件用来存储道路信息。

① 景点信息 ：景点编号、名字和介绍

编号	名字	介绍
0	A区	……
1	B区	……
2	C区	……
3	D区	……
4	E区	……
5	F区	……
6	G区	……

Vex.txt文件 第1行，记录景区的景点个数。 从第2行开始， 每3行记录一个景点信息。

文件格式与示例如下：

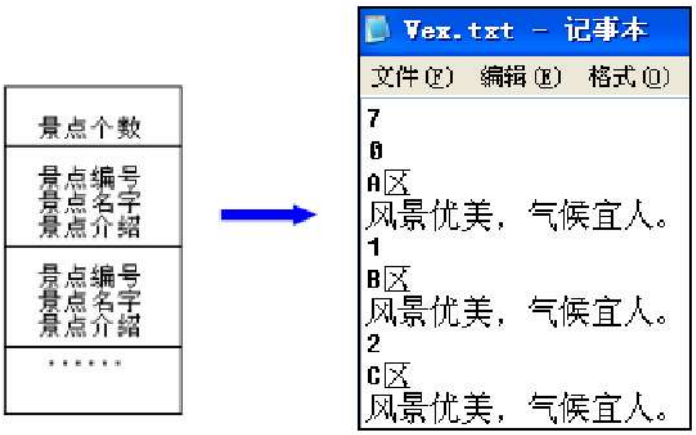


图6-2 Vex.txt文件

② 道路信息 ：景点1，景点2、两个景点之间的距离。

景点1	景点2	距离(m)

A	C	700
A	E	1000
A	F	600
B	C	1000
B	G	1000
C	D	
D	E	300
D	G	400
E	F	600
F	G	500

Edge.txt文件中， 每1行记录1条道路信息 。格式为：“ 景点1的编号 景点2的编号 道路的长度 ”。（每个字段使用空格符分割）

若文件中没有某两个景点的信息，就表示这两个景点之间没有直接的路径。

文件格式及示例如下：

图6-3 Edge.txt文件

（2）系统功能

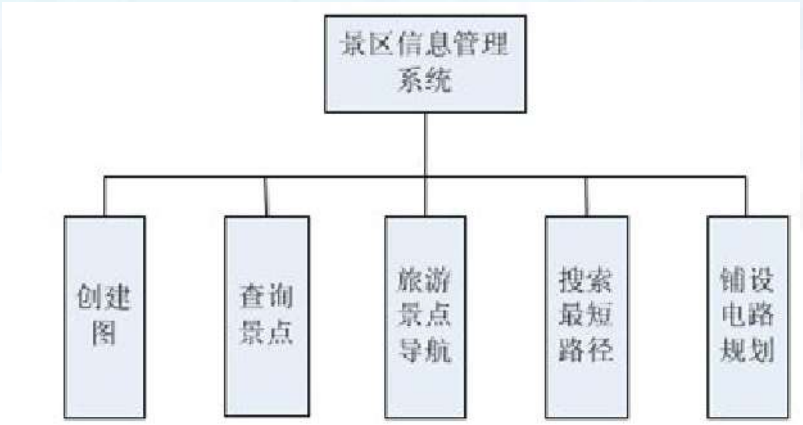


图6-4 功能结构图

1）创建图

输入：从Vex.txt文件中读取景点信息，从Edge.txt文件中读取道路信息。

处理：根据读取的景区信息创建景区景点图。

输出：

创建成功 ，依次输出：

① 顶点数目

② 顶点编号

③ 顶点名字

③ 边两端的顶点编号

⑤ 边的权值

创建失败 ，输出失败的提示信息。

输出格式如下：

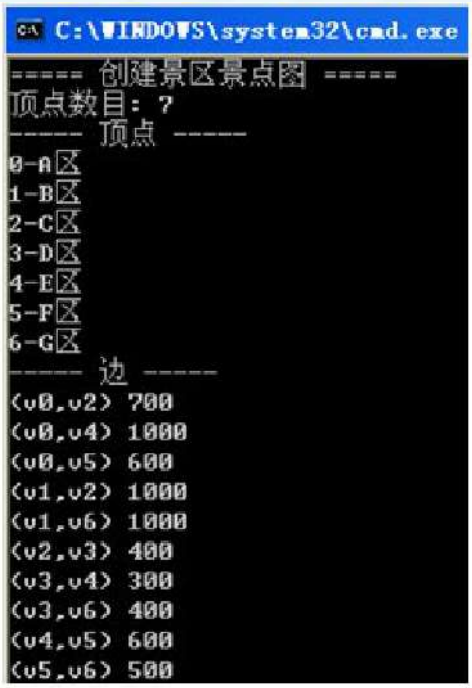


图6-5 图的信息

2）查询景点

输入：想要查询的景点的编号。

处理：根据输入的景点编号，查询该景点及相邻景点的信息。

输出：

① 景点名字

② 景点介绍

③ 相邻景区的名字

④ 到达相邻景区的路径 长度

输出格式如下：

图6-6 景点信息

3）旅游景点导航

输入：起始景点的编号。

处理：使用深度优先搜索(DFS)算法，查询以该景点为起点，无回路游览整个景区的路线。

输出：所有符合要求的导航路线。

输出格式如下：

图6-7 景点导航

4）搜索最短路径

输入：

① 起始景点的编号

② 终点的编号。

处理：使用迪杰斯特拉(Dijkstra)算法，求得从起始景点到终点之间的最短路径，计算路径总长度。

输出：

① 最短路线

② 路径总长度

输出格式如下：

图6-8 最短路径和距离

5）铺设电路规划

输入：景区的所有景点信息和道路信息

处理：根据景区景点图使用普里姆(Prim)算法构造最小生成树，设计出一套铺设线路最短，但能满足每个景点都能通电的方案。

输出：

① 需要铺设电路的道路

② 每条道路铺设电路的长度

③ 铺设电路的总长度

输出格式如下：

图6-9 铺设电路规划

2、程序分析和设计

程序输入的数据，可从文件中直接读取，也可以从界面中输入。本程序将直接从Vex.txt和Edge.txt这两个文件中读取数据，采用图的存储结构，应用深度优先搜索算法 (DFS) 、迪杰斯特拉(Dijkstra)算法、普里姆(Prim)算法等，开发景区信息管理系统。

（1）程序设计

使用Mircosoft Visual Studio 2010开发工具，创建一个空的控制台工程(Win32 Console Application)。利用图的存储结构来保存景区景点图，使用C++语言开发景区信息管理系统，工程名为 GraphCPro 。

① 添加 CGraph 类，用来定义图的数据结构，实现图的相关操作。

② 添加 CTourism 类，用来实现景区信息管理系统的相关功能。

③ 添加 Main.cpp 文件，在文件中创建程序入口函数 int main(void) 。

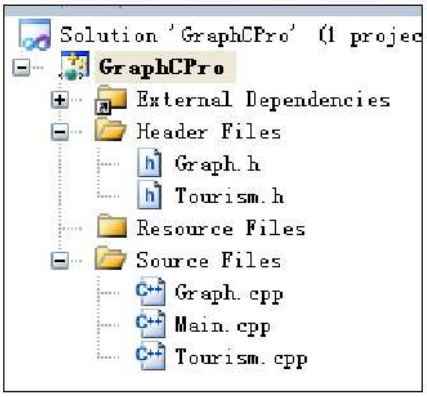


图6-10 工程结构

（2）界面设计

在int main(void)函数中输出菜单，将系统功能列出来，供用户选择。

输入数字	功能
1	创建景区景点图
2	查询景点信息
3	旅游景点导航
4	搜索最短路径
5	铺设电路规划
0	退出

使用while循环输出主菜单。使用cin获得用户的输入，使用switch-case语句判断具体是哪个功能。界面效果图如下：

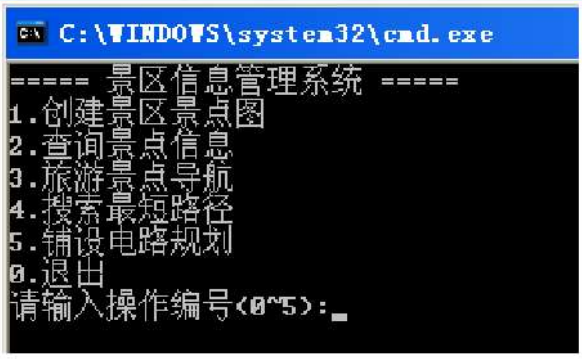


图6-11 界面设计

（3）算法设计

① 旅游景点导航

使用 深度优先搜索算法(DFS) 遍历景区景点图，得到一条导航路线；

改进深度优先搜索算法，得到多条导航路线。

② 搜索最短路径

使用 迪杰斯特拉(Dijkstra)算法 求得两个顶点间的最短路径。

③ 铺设电路规划

使用 普里姆(Prim)算法 构造最小生成树，得到铺设电路规划图。

（4）类设计

本程序将使用两个类：CGraph类与CTourism类。CGraph类用于实现图的数据结构和相关操作。CTourism类用于实现系统的主要功能。

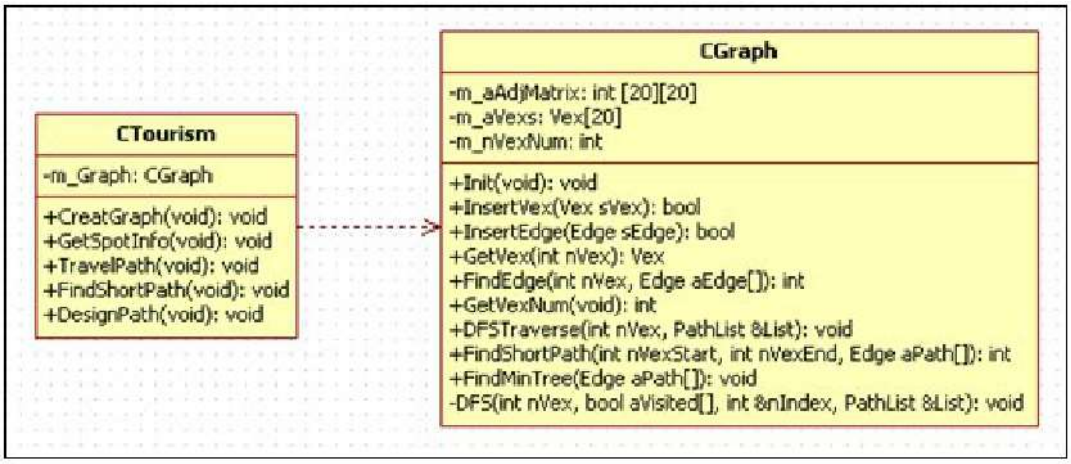


图6-12 类的定义和关系

① CGraph类

CGraph类为程序的数据结构类，用于存储景区景点图，实现图的相关操作。

② CTourism类

CTourism类为景点旅游信息管理功能类，用于实现系统的主要功能。

（5）数据结构设计

1）图的存储

当保存图结构时，即要保存 顶点信息 ，也要保存 边 。图可用数组或链表来存储。

① 数组表示 ，常用一维数组来保存顶点的集合，使用二维数组来保存边的集合；

② 链表表示 ，常用 邻接表 、 十字链表 等方式存储图的顶点和边的信息。

本程序中使用数组表示法存储图：

定义 一维数组 Vex m_aVexs[20] 保存顶点信息，最多允许有20个顶点。

定义 二维数组 （邻接矩阵） int m_AdjMatrix[20][20] 保存边的集合，数组中 每个元素的值即为边的权值 。

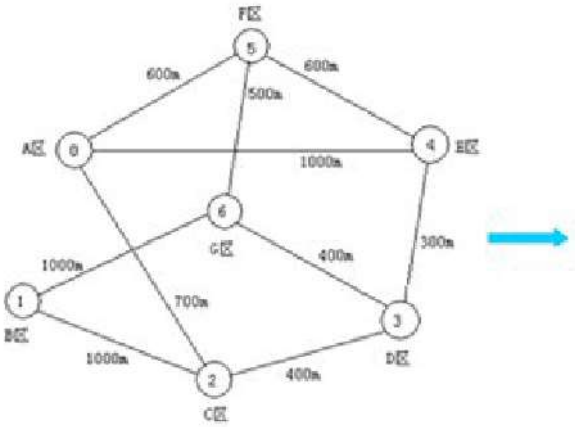
2）景区景点图

景区的地图可以看做是一个 带权无向图 ，使用 邻接矩阵 来保存。

① 景区中的 所有景点即为图的 顶点 。

② 当两个景点之间铺设的 有道路时，表示两个顶点相连，为一条 边 。

③ 两个景点之间的距离，即为边的 权值 。权值为0表示两个顶点不相连。



	0	1	2	3	4	5	6
0	0	0	700	0	1000	600	0
1	0	0	1000	0	0	0	1000
2	700	1000	0	400	0	0	0
3	0	0	400	0	300	0	400
4	1000	0	0	300	0	600	0
5	600	0	0	0	600	0	500
6	0	1000	0	400	0	500	0

图6-13 景区景点图

3）顶点和边的信息

① 定义Vex结构体，存储图的顶点信息。

```
struct Vex
{
int num; // 景点编号
char name[20]; // 景点名字
char desc[1024]; // 景点介绍
};
```

② 定义Edge结构体，存储图的边的信息。

```
struct Edge
{
int vex1; //边的第一个顶点
int vex2; //边的第二个顶点
int weight; //权值
};
```

3、开发实现

参考“景区信息管理系统”。