

1 Dijkstra 算法与搜索最短路径实践

1.1 实验目标

- 1、掌握迪杰斯特拉(Dijkstra)算法和应用。
- 2、基于已经实现“景区信息管理系统”功能，采用迭代开发，使用 C++语言和 Dijkstra 算法实现“搜索最短路径”功能开发。

1.2 实验任务

在“旅游景点导航”工程的基础上，为景区信息管理系统增加“搜索最短路径”功能。

- (1) 输入
 - 1) 起点编号 int nVexStart;
 - 2) 终点编号 int nVexEnd。
- (2) 处理
 - 搜索两个景点之间的所有路径，找到其中距离最短的路径。
- (3) 输出
 - 在控制台输出最短路径及路径的总长度。



图 1-1 搜索最短路径

1.3 分析和设计

在“创建图和查询景点”或“旅游景点导航”工程的基础上进行迭代开发。

- 1、算法设计
 - 常用的求最短路径的算法有两个:
 - (1) 迪杰斯特拉(Dijkstra)算法

特点：求得从某个源点到其余各顶点的最短路径

(2) 弗洛伊德(Floyd)算法

特点：求得每一对顶点之间的最短路径

本程序首先使用 Dijkstra 算法来求得从某个起点到其余各顶点之间的最短路径，然后从中选出我们需要的那条。

对于同一顶点上的不同邻接点，按照其存储顺序进行访问。

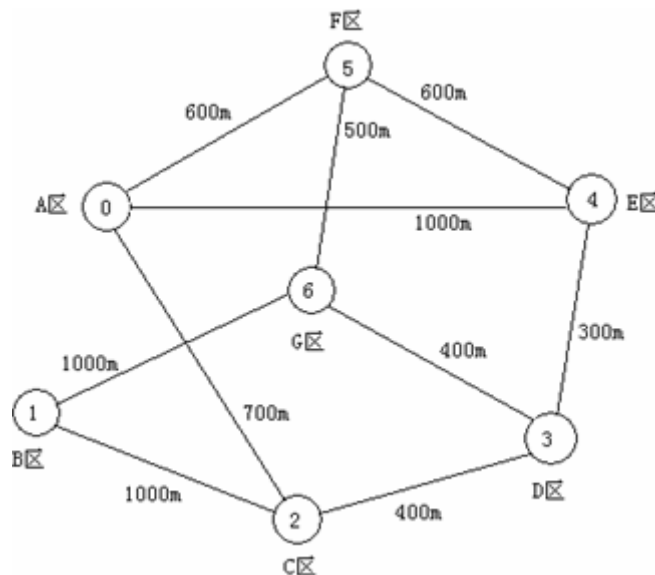


图 1-2 景区图

2、算法设计

(1) Graph.cpp 文件

增加函数：int FindShortPath(int nVexStart, int nVexEnd, Edge aPath[])

输入：起始景点的编号 v1 和目的景点的编号 v2。

输出：最短路径。

功能：通过 Dijkstra 算法求得 v1 到 v2 的最短路径。

(2) Tourism.cpp 文件

增加成员函数：void FindShortPath(void)

功能：通过调用 m_Graph() 函数查询两个景点间的最短路径和距离。

1.4 编码实现

为景区信息管理系统增加搜索最短路径功能。使用 Dijkstra 算法求得两个顶点间的最短路径和最短距离，并显示在界面上。

在实现过程中采用迭代思路，具体实现步骤如下：

步骤一：导入工程。

步骤二：搜索最短路径。

步骤三：查询最短路径。

步骤四：编译和运行。

1.4.1 导入工程

在实现了旅游景点导航的功能之后，要在原有工程的基础上新增搜索最短路径功能。

- (1) 打开 Microsoft Visual Studio 2010 开发工具。
- (2) 导入“GraphCPro”工程。

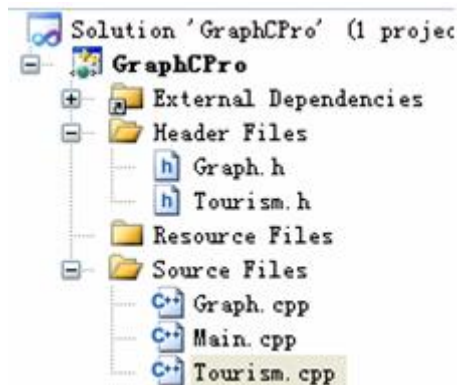


图 1-3 程序结构

1.4.2 搜索最短路径

使用 Dijkstra 算法搜索最短路径，对于同一顶点上的多个邻接点，按照它们的存储顺序来访问。

在 Graph.cpp 文件中添加 **FindShortPath()**方法，通过 Dijkstra 算法搜索源点到终点之间的最短路径。

```
int FindShortPath(int nVexStart,int nVexEnd,Edge aPath[])
{
    //初始化最短路径
    for(...;...;...){
        if(w 是距离 nVexStart 最近的点&&!aVisited[w]){
            aVisited[w] = true
        }
        // 将 w 作为中间点计算 nVexStart 到所有顶点的最短距离
        if(有新的最短距离){
            //更新最短路径
        }
    }
}
```

1.4.3 查询最短路径

在 Tourism.cpp 文件中添加 **FindShortPath()**方法，通过调用 FindShortPath 函数，查询起

始景点到目的景点之间的的最短路径和最短距离。

```
void FindShortPath(void)
{
    //输入两个景点的编号
    //搜索最短路径
    //得到最短路径和最短距离
}
```

1.5 调试和运行

(1) 按 Ctrl+F5，编译运行程序。

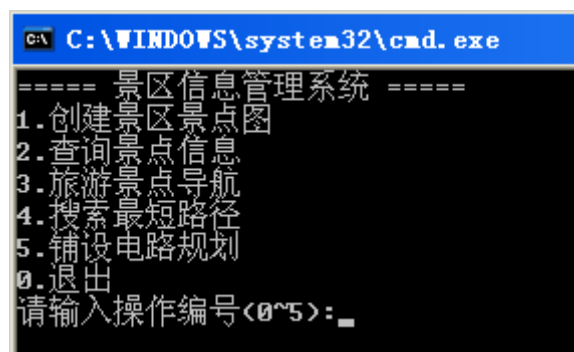


图 1-4 主菜单

(2) 在主菜单中输入 4，则进行对应“旅游景点导航”操作。

在 main()函数 case 4 语句中调用 FindShortPath()函数，进入旅游景点导航功能。编译和运行，得到结果：



图 1-5 运行结果

2 最小生成树与铺设电路规划实践

2.1 实验目标

- 1、理解最小生成树的概念。
- 2、掌握普里姆(Prim)算法和应用。
- 3、基于已经实现“景区信息管理系统”功能，采用迭代开发，使用 C++ 语言和最小生成树算法实现“铺设电路规划”功能开发。

2.2 实验任务

在“搜索最短路径”工程的基础上，为景区信息管理系统增加“铺设电路规划”的功能。

(1) 输入

景区景点信息和道路信息。

(2) 处理

根据景区景点图，构造一棵最小生成树，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。

(3) 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。

2.3 分析和设计

在“创建图和查询景点”或“搜索最短路径”工程的基础上进行迭代开发。

1、算法设计

铺设电路的规划方案要求铺设线路最短，但能满足每个景点都能通电，从图的角度来说，就是在图的所有路径中，选择其中的几条，使得这几条路径可以连通所有的顶点，并且权值和最小。

得到这几条路径的过程，实际上就是构造最小生成树的过程。

构建最小生成树常用的方法有两种，普里姆(Prim)算法和克鲁斯卡尔(Kruskal)算法。本程序使用 Prim 算法来构建最小生成树。

对于下边的图，以顶点 0 为源点，按照顶点的存储顺序查找顶点，构建最小生成树。

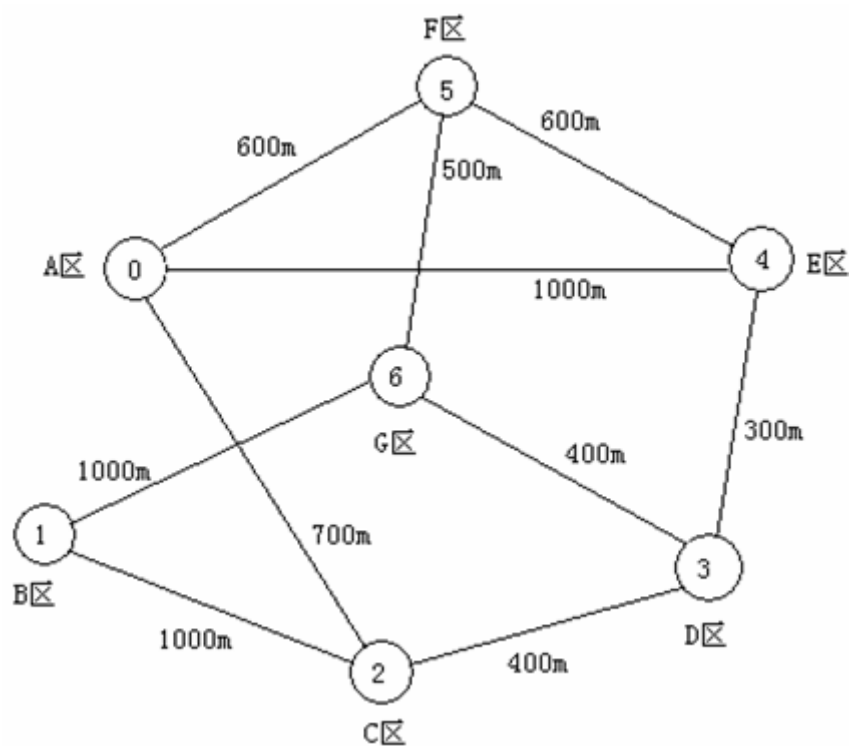


图 2-1 原图

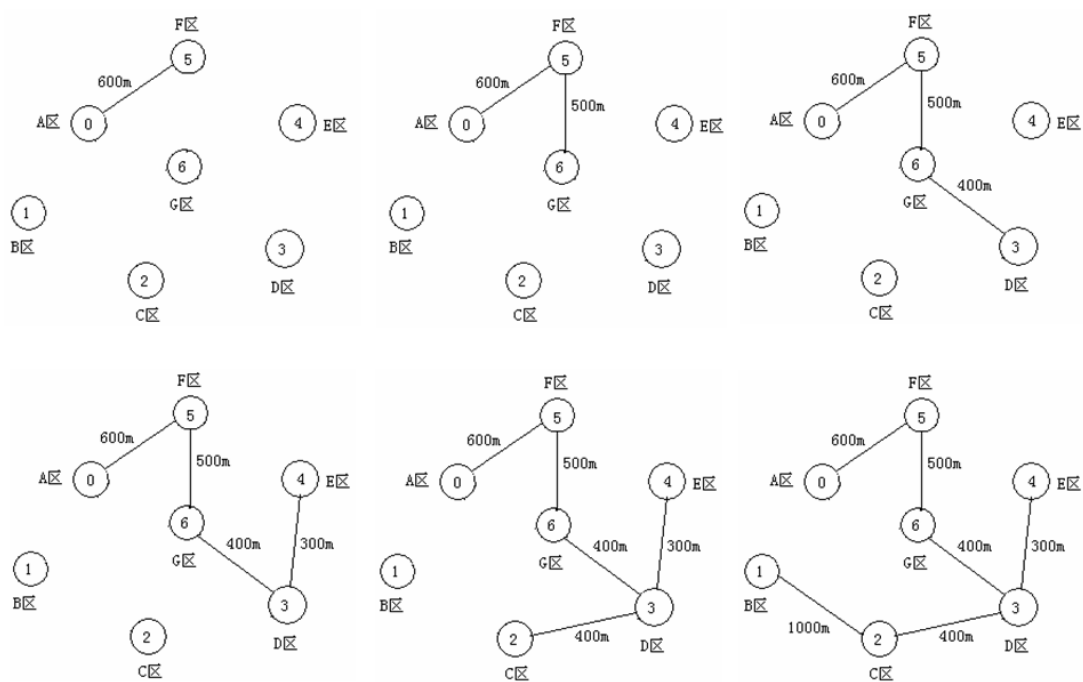


图 2-2 最小生成树

查找 v1 和 v2 的顺序不同，得到的最小生成树可能会不同。

下面是按照顶点的存储顺序查找和逆序查找得到的不同的最小生成树：

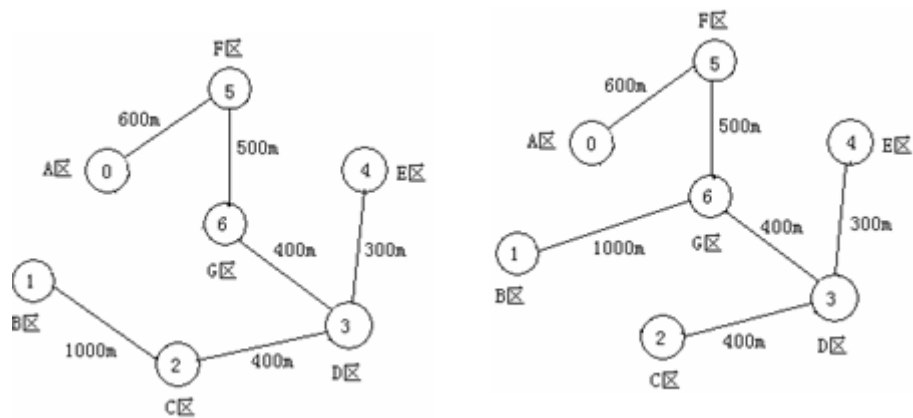


图 2-3 不同的最小生成树

图的结构决定了最小生成树的个数，相同权值的边越多，最小生成树的个数就可能越多。

对于同一个图，得到多个最小生成树的方式有两种：

- (1) 改变顶点的查找顺序
- (2) 当有多条权值最小的边时，使用栈将每条边都保存下来

2、类的设计

(1) Graph.cpp 文件

增加函数：int FindMinTree(Edge aPath[])

输入：Edge aPath[]

输出：最小生成树。

功能：通过 Prim 算法构建最小生成树。

(2) Tourism.cpp 文件

增加函数：void DesignPath(void)

输入：viod

输出：铺设的线路。

功能：通过调用 Graph.cpp 文件中的 FindMinTree()方法查询铺设电路规划图。

3、最小生成树

在一给定的无向图 $G = (V, \{E\})$ 中， (u, v) 代表连接顶点 u 与顶点 v 的边，而 $w(u, v)$ 代表此边的权重，若存在 T 为 E 的子集且为无循环图，使得 $w(T)$ 最小，则此 T 为 G 的最小生成树。

4、Prim 算法

假设 $N = \{V, \{E\}$ 是连通图 $\}$ ， TE 和 TV 分别是最小生成树的边和顶点的集合，以 v_0 作为源点，构造最小生成树：

- (1) 将 v_0 从 V 移动到 TV 中；
- (2) 在 V 中找到顶点 v_1 ， TV 中找到顶点 v_2 ，使得边 $E(v_1, v_2)$ 的权值最小；
- (3) 将 $E(v_1, v_2)$ 加入到 TE 中， v_1 移动到 TV 中，重复步骤 2，直到 V 为空，此时 TE 即为 N 的最小生成树。

2.4 编码实现

为景区信息管理系统增加铺设电路规划功能。当用户选择铺设电路规划功能时，系统会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。

在实现过程中采用迭代思路，具体实现步骤如下：

步骤一：导入工程

步骤二：构建最小生成树

步骤三：查询铺设电路规划图

步骤四：编译和运行

2.4.1 导入工程

在实现了搜索最短路径的功能之后，在原有工程的基础上新增铺设电路规划功能。

(1) 打开 Microsoft Visual Studio 2010 开发工具。

(2) 导入“GraphCPro”工程。

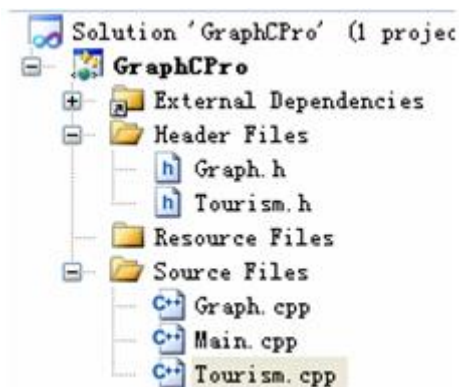


图 2-4 导入工程

2.4.2 构建最小生成树

使用 Prim 算法构建最小生成树，按照顶点的存储顺序来查找顶点。

在 Graph.cpp 文件中增加 FindMinTree() 方法，通过普里姆算法构建最小生成树。

```
int FindMinTree(Edge aPrth[])
{
    //使用 Prim 算法构建最小生成树
}
```

2.4.3 查询铺设电路规划图

在 Tourism.cpp 文件中增加 DesignPath () 方法，通过调用 FindMinTree () 方法查询铺设电

路规划图。

```
int DesignPath()
{
    M_Graph.Find MinTree() //构建最小树

    //输出铺设线路图
}
```

2.5 调试和运行

(1) 按 Ctrl+F5，编译和运行程序。

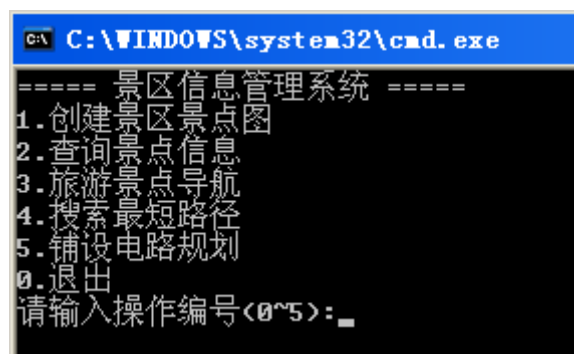


图 2-5 主菜单

(2) 在主菜单中输入 5，则进行对应“铺设电路规划”操作。

在 main() 函数 case 5 语句中调用 DesignPath() 方法，进入铺设电路规划功能。编译和运行，得到结果：



图 2-6 运行结果