最短校园路径规划系统设计

**项目组号：4**

**组员：杨婧蕙 孔雯 李闽榕**

**日期：2022年4月23日**

**目录**

**[1.背景 2](#_Toc8970)**

**[2.需求分析 3](#_Toc27363)**

**[3. 页面简明介绍 4](#_Toc1049)**

**[3.1 目前程序完成功能 4](#_Toc18792)**

**[3.2 用户界面设计 4](#_Toc23826)**

**[3.3 预期完成的功能 4](#_Toc29695)**

**[4 数据结构设计 5](#_Toc875)**

**[4.1 Vertex 类（存储地点信息） 5](#_Toc15039)**

**[4.2 路径维护 5](#_Toc32108)**

**[4.3 方法 5](#_Toc28440)**

**[4.4 详细设计 5](#_Toc20967)**

**[5.主题算法详解 7](#_Toc4915)**

**[5.1 基本思想 7](#_Toc7831)**

**[5.2 操作步骤 8](#_Toc10565)**

**[5.3 图解 8](#_Toc29339)**

**[6.算法优势介绍 9](#_Toc19117)**

**[7.使用的语言（技术）及其优势 10](#_Toc5544)**

**[7.1程序开发运行选用的环境 ： 10](#_Toc11477)**

**[7.2程序开发使用的语言：c++ 10](#_Toc27109)**

**1.背景**

为帮助新生和校外访客更快地了解学校道路、建筑布局，为避免教职工和学生绕远路而浪费过多时间，本组研发了简易的校园路径规划系统，以减少不必要的时间损耗，提高时间利用率。同时，随着移动通信技术的不断发展，“数字校园”的概念应运而生，通过启用图形、图像数字信息表现校园的各种空间及属性要素，为用户提供各种校园信息的查询、检索和必要的空间分析、统计操作以及按不同的用户要求输出相应的专题要素，为校园的发展预测、规划决策以及科学管理提供可靠依据。本系统也进一步加强了我校数字化校园建设，具有深远的现实意义。

**2.需求分析**

旅行商问题：

一个售货员必须访问n个城市，恰好访问每个城市一次，并最终回到出发城市。售货员从城市i到城市j的旅行费用是一个整数，旅行所需的全部费用是他旅行经过的各边费用之和，而售货员希望使整个旅行费用最低。（等价于求图的最短哈密尔顿回路问题）令 G=(V, E)是一个带权重的有向图，顶点集V=(v0, v1, ..., vn-1)。从图中任一顶点vi出发，经图中所有其他顶点一次且只有一次，最后回到同一顶点vi的最短路径。

算法分析：

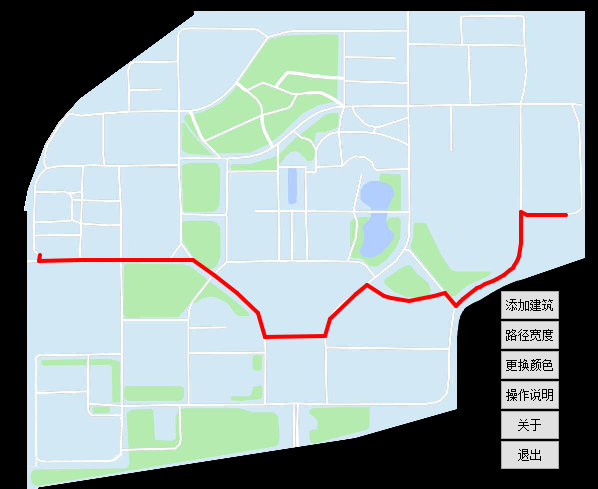
在设计中，需要考虑时间复杂度及空间复杂度，考虑用户的路径可能存在转弯、圆圈形状。若直接以两点直线距离作为路径则不合规范且估算不符合现实情况，所以在设计的过程中需要考虑到路径内的拐点，在每个路口单独设计位置信息点，这样才能模拟出实际用户行走线路。同时，直线可直接用两端点为信息坐标存储。页面设计方面，需要考虑用户操作复杂度，用户直接点击起始点位置以减少操作复杂度，在附加功能设计时，使用 push button 使用户仅需要点击按钮即可实现功能，无需输入过多信息。

1. **页面简明介绍**

**3.1 目前程序完成功能**

* 展示校园平面图，供用户选择性点击出发点终止点；
* 按用户选择，在校园平面图上提供合理出行线路和相关信息；
* 附加功能：地图变换（滚轮放大缩小）
* 附加功能：地图变换（右键拖动位置）
* 附加功能：路径粗细变换
* 附加功能：路径颜色变换

**3.2 用户界面设计**



**3.3 预期完成的功能**

页面增加“导入地图”的按钮，可使系统自动识别地图并生成相应解决方案，最终实现对任意地图的导航。

**4 数据结构设计**

**4.1 Vertex 类（存储地点信息）**

**4.2 路径维护**

**4.2.1 使用泛型集合来存储维护路径**

**4.3 方法**

void didianzuobiao(); //导入地点信息

void guanjiandianzuobiao(); //导入位置点信息

void biandejianli(); //导入边信息

void julijuzhen(); //导入邻接矩阵信息

void jisuanZuiduanLujing(int s); //计算最短路径

void xinxijianli(); //导入景点信息

void TSP(); //TSP 算法

bool isVisited(bool visited[]);

void printPath();

void getPath();

void hamilton(); //生成 Hamilton 回路

**4.4 详细设计**

### 4.4.1 信息导入

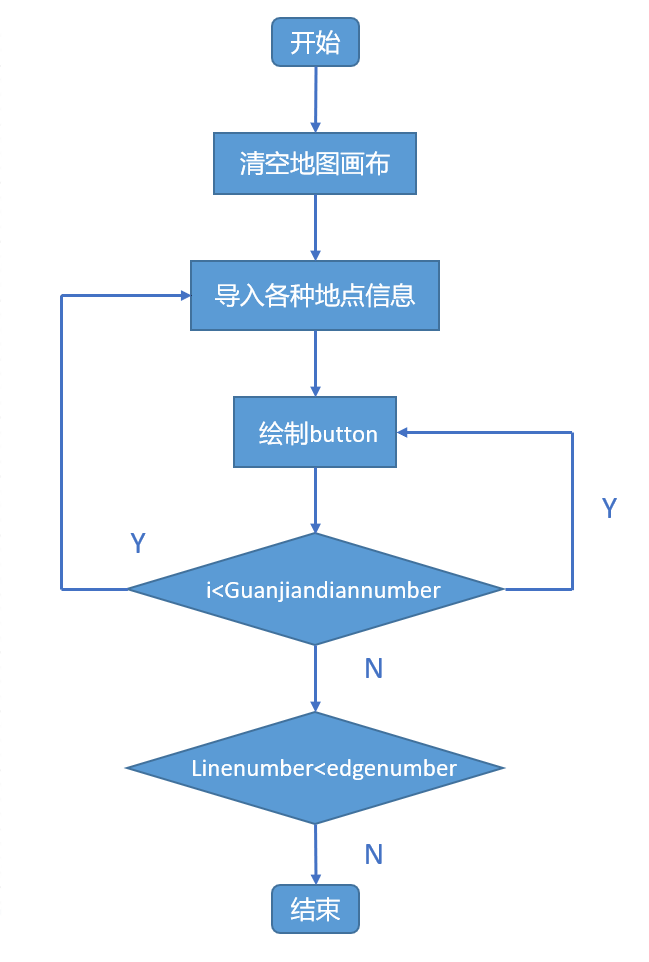
void didianzuobiao(); //导入地点信息

void guanjiandianzuobiao(); //导入位置点信息

void biandejianli(); //导入边信息

void julijuzhen(); //导入邻接矩阵信息

### 框图

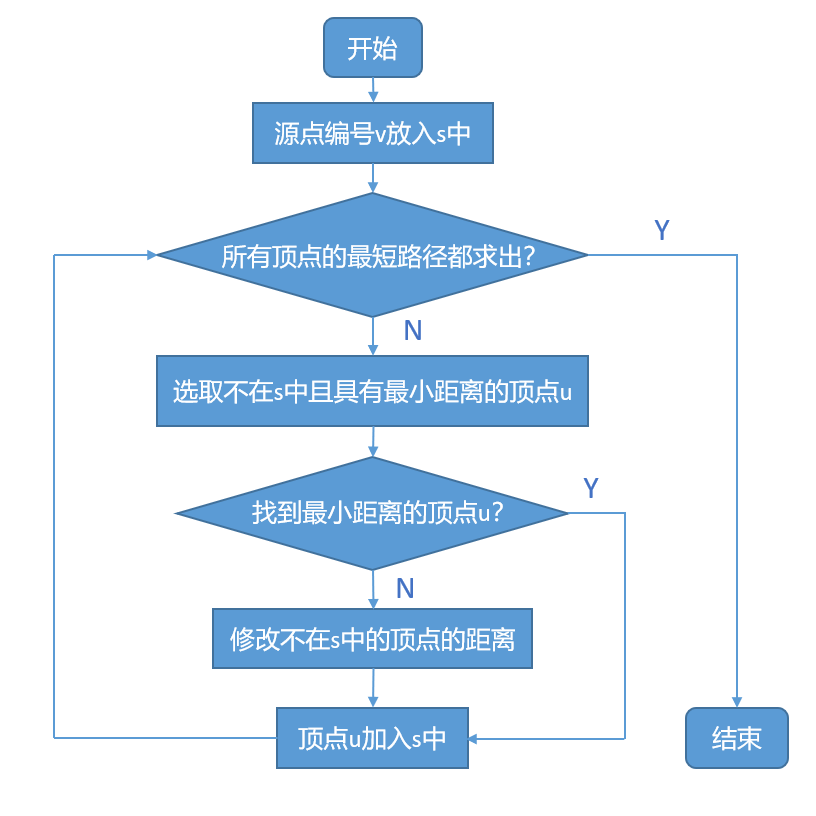
****

### 4.4.2 距离矩阵的建立

### 4.4.2.1使用功能

选择要去的地点，依次选择起点及中途点，使用 Dijkstra 算法，求出最短距离矩阵，再在每个距离路径已知的情况下，组成路径根据起点及中途点查询最短路径，并用红线标出。

### 4.4.2.2迪杰斯特拉算法



**5.主题算法详解**

迪杰斯特拉(Dijkstra)算法是典型[最短路径](https://so.csdn.net/so/search?q=%E6%9C%80%E7%9F%AD%E8%B7%AF%E5%BE%84&spm=1001.2101.3001.7020)算法，用于计算一个节点到其他节点的最短路径。它的主要特点是以起始点为中心向外层层扩展([广度优先搜索](https://so.csdn.net/so/search?q=%E5%B9%BF%E5%BA%A6%E4%BC%98%E5%85%88%E6%90%9C%E7%B4%A2&spm=1001.2101.3001.7020)思想)，直到扩展到终点为止。

**5.1 基本思想**

通过Dijkstra计算图G中的最短路径时，需要指定起点s(即从顶点s开始计算)。

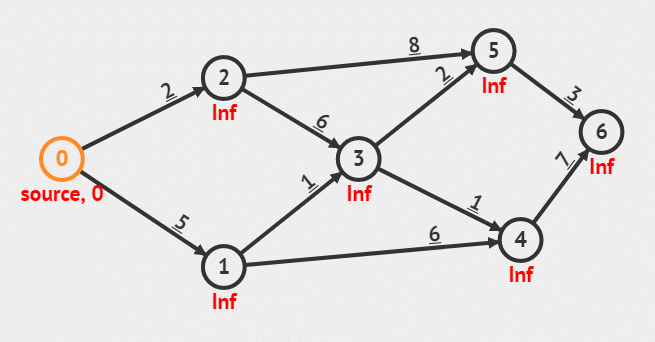
此外，引进两个集合S和U。S的作用是记录已求出最短路径的顶点(以及相应的最短路径长度)，而U则是记录还未求出最短路径的顶点(以及该顶点到起点s的距离)。

初始时，S中只有起点s；U中是除s之外的顶点，并且U中顶点的路径是“起点s到该顶点的路径”。然后，从U中找出路径最短的顶点，并将其加入到S中；接着，更新U中的顶点和顶点对应的路径。 然后，再从U中找出路径最短的顶点，并将其加入到S中；接着，更新U中的顶点和顶点对应的路径。 … 重复该操作，直到遍历完所有顶点。

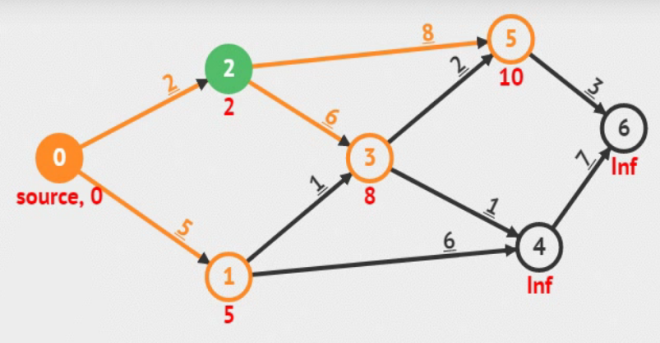
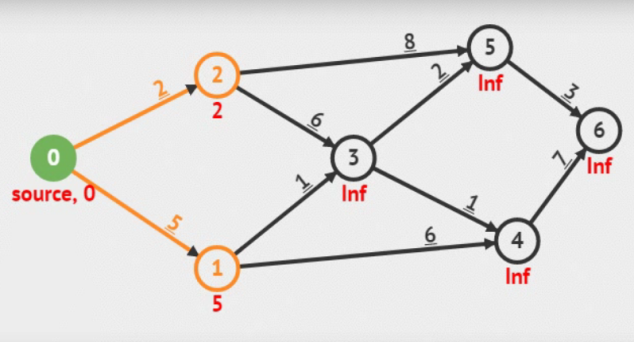
**5.2 操作步骤**

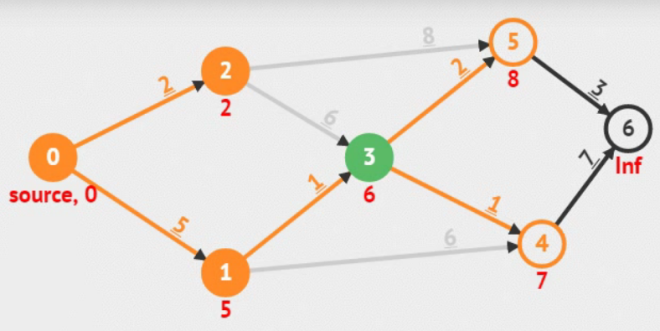
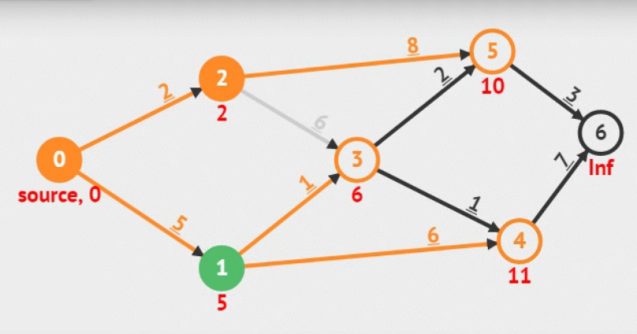
1. 初始时，S只包含起点s；U包含除s外的其他顶点，且U中顶点的距离为“起点s到该顶点的距离”[例如，U中顶点v的距离为(s,v)的长度，然后s和v不相邻，则v的距离为∞]。
2. 从U中选出“距离最短的顶点k”，并将顶点k加入到S中；同时，从U中移除顶点k。
3. 更新U中各个顶点到起点s的距离。之所以更新U中顶点的距离，是由于上一步中确定了k是求出最短路径的顶点，从而可以利用k来更新其它顶点的距离；例如，(s,v)的距离可能大于(s,k)+(k,v)的距离。
4. 重复步骤(2)和(3)，直到遍历完所有顶点。
5. 单纯的看上面的理论可能比较难以理解，下面通过实例来对该算法进行说明。

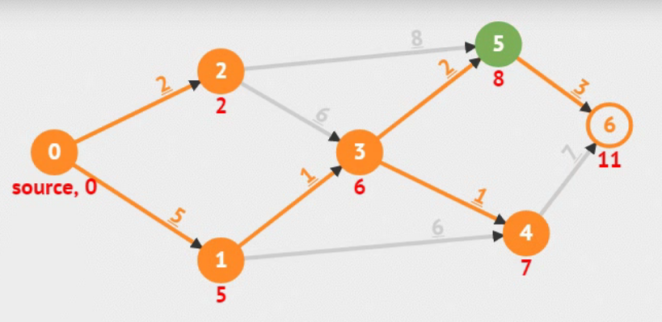
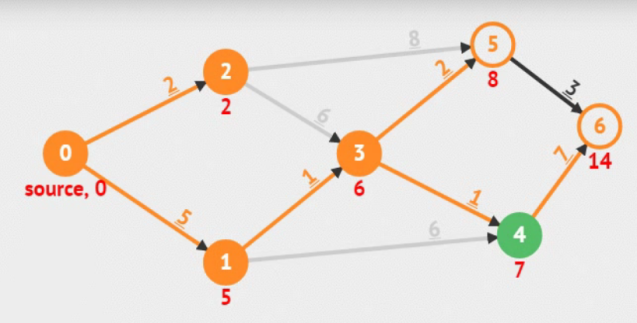
**5.3 图解**

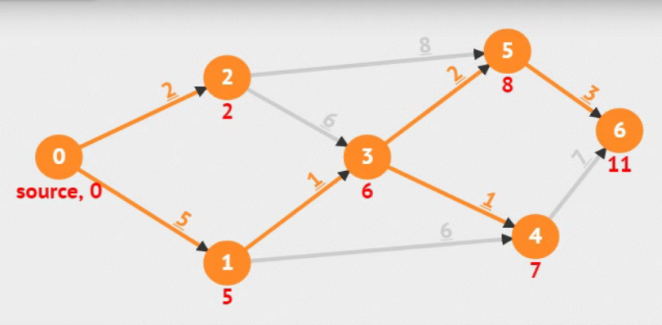
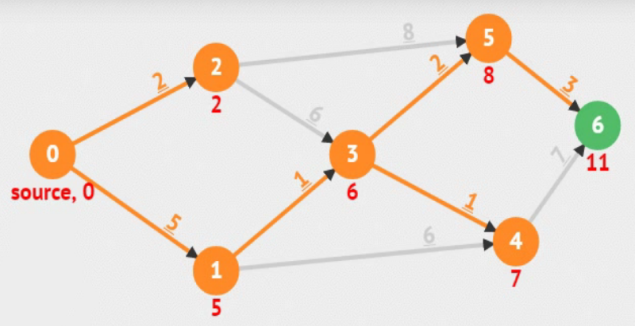
****

以上图为例，来对迪杰斯特拉进行算法演示(以0为起点)。









**6.算法优势介绍**

Dijkstra算法也被称为狄克斯特拉算法，是典型的用来解决最短路径的算法，也是很多教程中的范例，是由一个名为狄克斯特拉的荷兰科学家提出的，这种算法是计算从一个顶点到其他各个顶点的最短路径，虽然看上去很抽象，但是在实际生活中应用非常广泛，比如在网络中寻找路由器的最短路径就是通过该种算法实现的。在有向图比较复杂的情况下，相比其他求最短路径算法，该算法具有执行效率高，速度快等优点。大大减少了算法的时间复杂度，可表示为O(N\*N)，算法简明，且能得到最优解。

**7.使用的语言（技术）及其优势**

**7.1程序开发运行选用的环境 ：**

QT Creator 5.9.0（Community）

**7.1.1 QT优势：**

1.跨平台特性，避免重复造车轮，同时Qt不强制使用任何设计模式

2.QT本身优良的机制，使用面向组件编程。

3.把精力放在核心功能的实现上，而不是源码本身。

4.能构造出艺术级的界面，设计界面方便。

5.Qt拥有非常简单而又不失强大的layout机制，布局灵活多变

**7.2程序开发使用的语言：c++**

**7.2.2 c++的优势：**

1.C++ 是一种静态类型的、编译式的、通用的、大小写敏感的、不规则的编程语言，支持过程化编程、面向对象编程和泛型编程。

2.C++ 被认为是一种中级语言，它综合了高级语言和低级语言的特点，初学者易于上手。

4.C++ 是 C语言的一个超集，事实上，任何合法的 C 程序都是合法的 C++ 程序。

5. C++实现了面向对象程序设计。在高级语言当中，处理运行速度是最快的，大部分的游戏软件，系统都是由C++来编写的。

6. C++语言非常灵活，功能非常强大。如果说C语言的优点是指针，那么C++的优点就是性能和类层次结构的设计。

7.C++非常严谨、精确和数理化，标准定义很细致。

8. C++语言的语法思路层次分明、相呼应;语法结构是显式的、明确的。

1. **总结与展望**

本系统可提供校园内到达两点的最短路径，切实便捷了校内人员的日常生活，可被人们日常广泛应用。在未来，本系统预期实现导入任意符合要求的图片并生成对应解决方案，使用户能够自行导入相应地图并获取所需路线规划，不局限于对特定地区的路线处理，可应用于旅游观光、外卖送达等场景，前景大好。