**数据结构课程设计总结**

****

**学 号 1750844**

**姓 名 周展田**

**专 业 计算机科学与技术**

2019年7月

**一、 算法实现设计说明**

**1.1 题目**

给定一个图，完成：  
（1）建立并显示出它的邻接链表；

（2）分别用Prim算法和Kruskal算法构造其最小生成树，随时显示其构造的过程；

（3）给出某一确定顶点到其他顶点的最短路径；

（4）给出每一对顶点之间的最短路径。

**1.2软件功能**

可以输入具有若干个顶点和若干条边的无向网，或使用默认的无向网，完成邻接链表的建立和显示，完成Prim算法和Kruskal算法构造最小生成树的动态演示，并且根据输入的顶点构造输入顶点与其他顶点的最短路径，最后完成每一对顶点之间的最短路径的构造。

**1.3设计思想**

根据输入的无向网的顶点数边数和每条边的权值，构造邻接矩阵与邻接链表，并将邻接链表遍历显示。Prim算法从点出发构造最小生成树，Kruskal算法从最小边出发构造最小生成树，构造最小生成树时使用邻接矩阵较为方便，所以在构造无向网时同时构造邻接链表和邻接矩阵。求某一顶点到其他顶点的最短路径使用迪杰斯特拉算法实现，求每一对顶点之间的最短距离使用弗洛伊德算法完成，这两种最短距离同样借助邻接矩阵完成。

设计一个Graph类，包含顶点、边、邻接矩阵、邻接链表等属性，将相关的建立、邻接链表的显示、Prim、Kruskal、最短路径的实现函数作为其成员函数，在类内完成相关操作。

使用Qt实现此GUI程序，以QWidget类作为程序的基类，实现每个步骤的展示通过实例化不同的QDialog并添加相关控件完成，其中较为复杂的Prim和Kruskal算法的动态展示，通过设计继承QWidget的类，在其中使用QPainter绘制完成。

**1.4逻辑结构与物理结构**

所用逻辑结构为图形结构（无向网），所用物理结构有顺序存储结构（二维数组），也有随机存储的链式结构（邻接链表）。



**1.5开发平台**

使用C++语言编程，使用Visual Studio2019，Qt5.13.0 GUI库以及Qt Creator4.9.2开发平台。

软件运行的环境：Windows 10 系统

**1.6系统的运行结果分析说明**

（1）开发过程

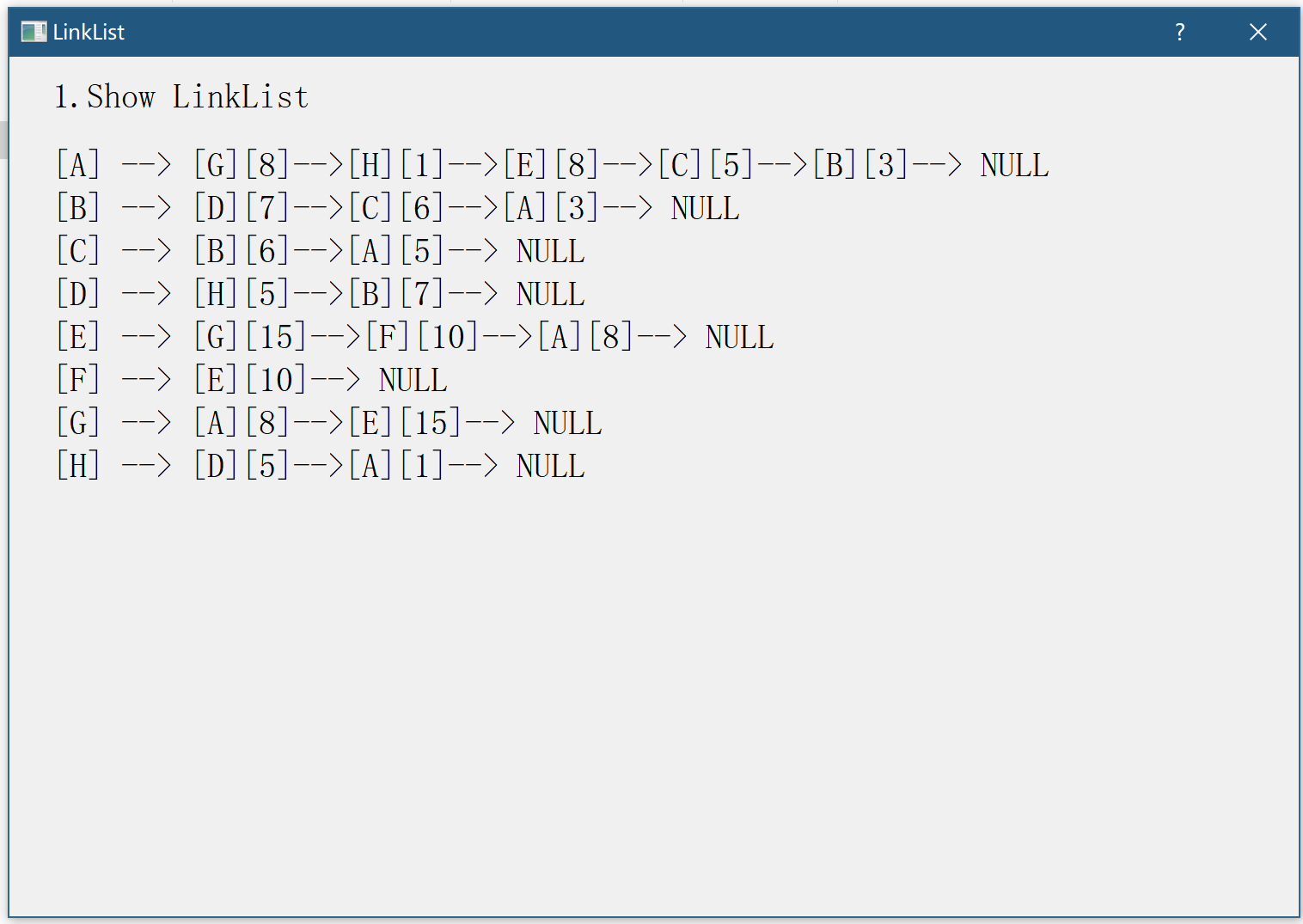
首先使用Visual Studio2019编写控制台程序，实现控制台界面的基本功能，包括链表的建立，最小生成树的构造和最短路径的实现，并进行一系列测试，在保证功能实现后，使用Qt进行GUI开发。

程序使用了MSVC 2017 64bit, MinGW32和MinGW64三种编译器进行了编译调试，其中MSVC 2017 64bit在进行构建时会出现部分中文输出无法编码的问题，将其内容更改后解决。使用三种编译器均通过了编译调试。

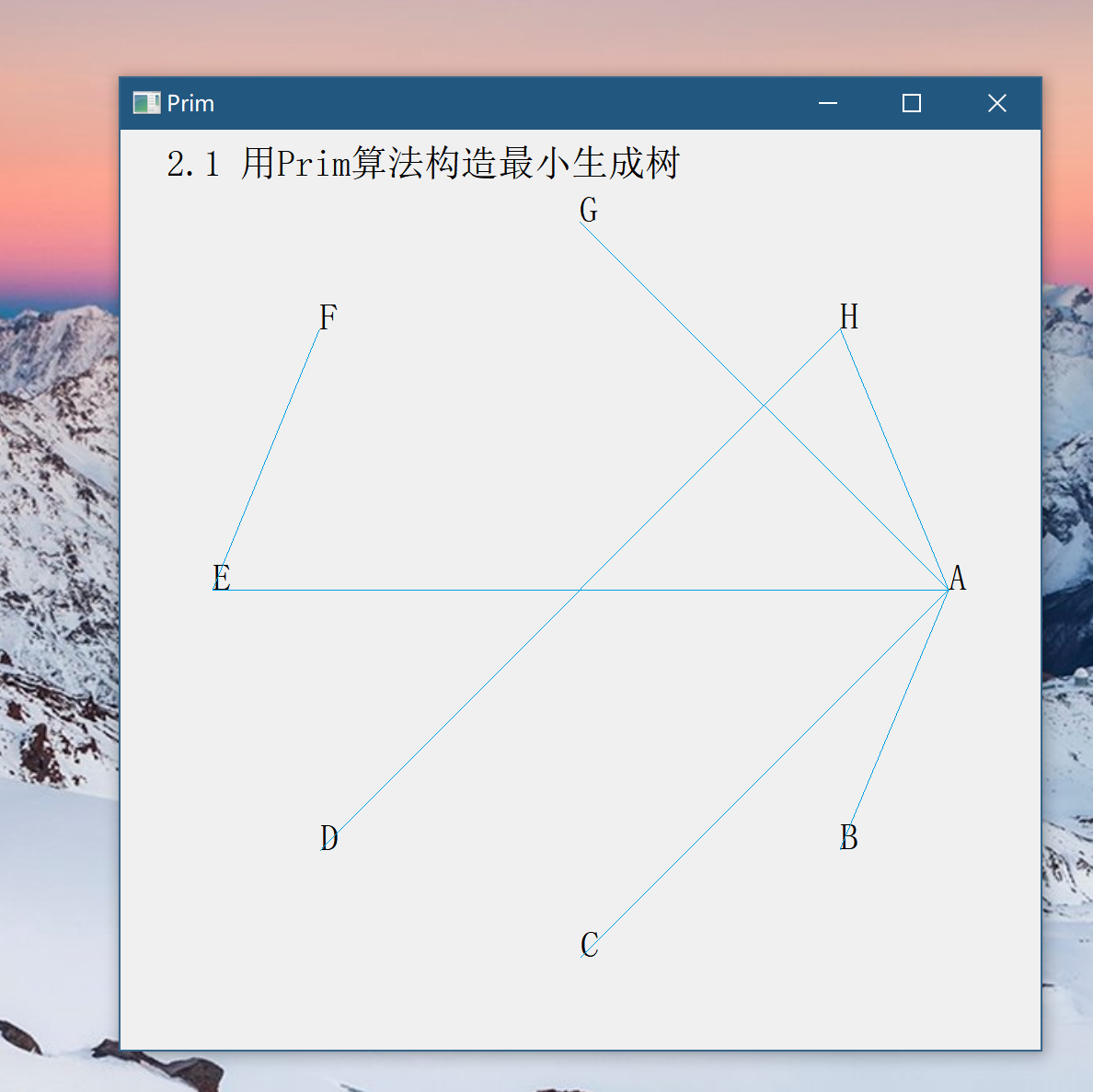
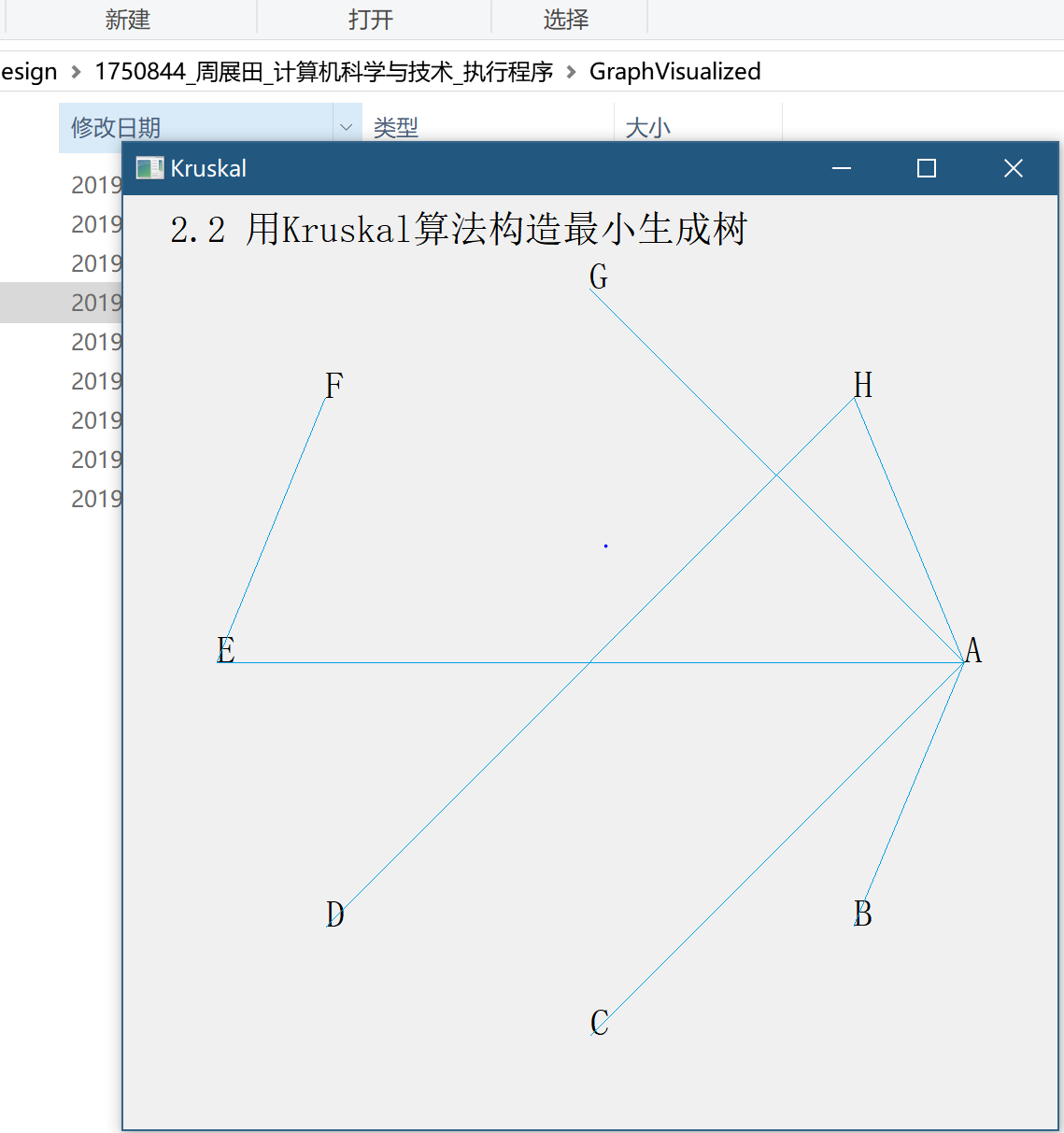
（2）达到成果：

正确性：

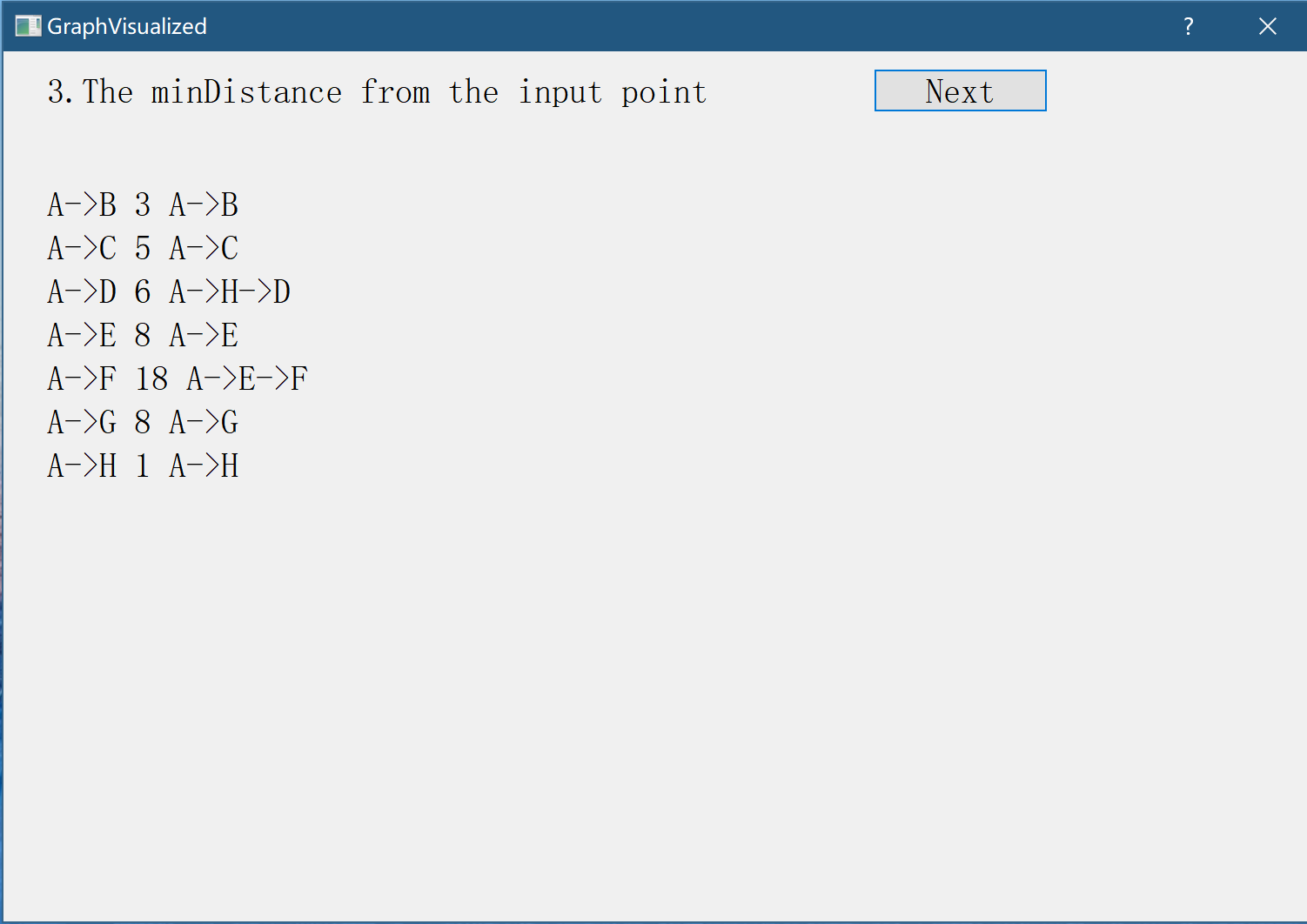
可以正确地显示出邻接链表



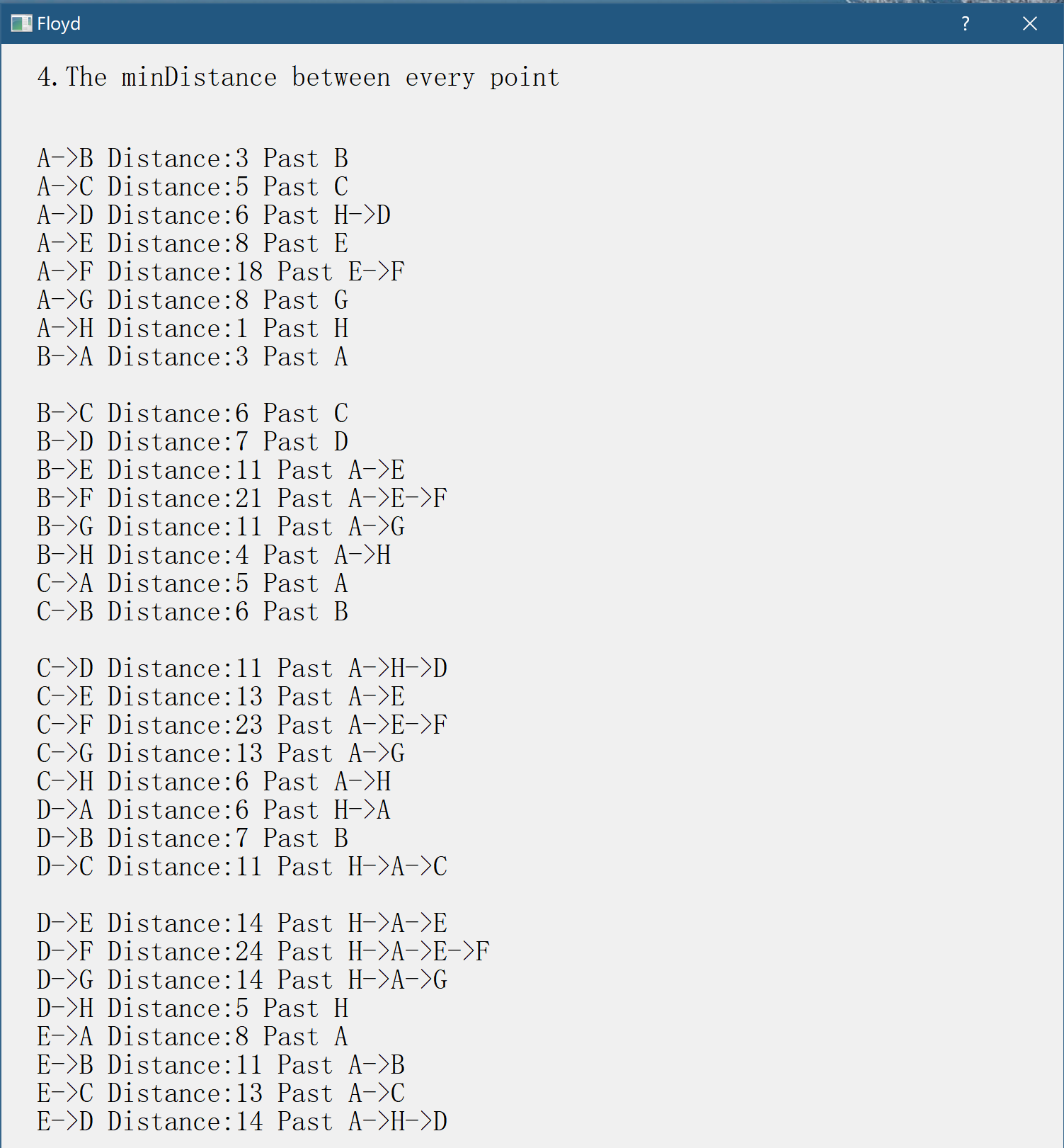
可以正确使用Prim算法和Kruskal算法动态生成最小生成树

可以正确的构造从某一点出发到其他顶点的最短路径，输出最短路径的长度以及所经过的顶点



可以正确构造每一对顶点之间的最短路径，输出最短路径的长度和经过的顶点

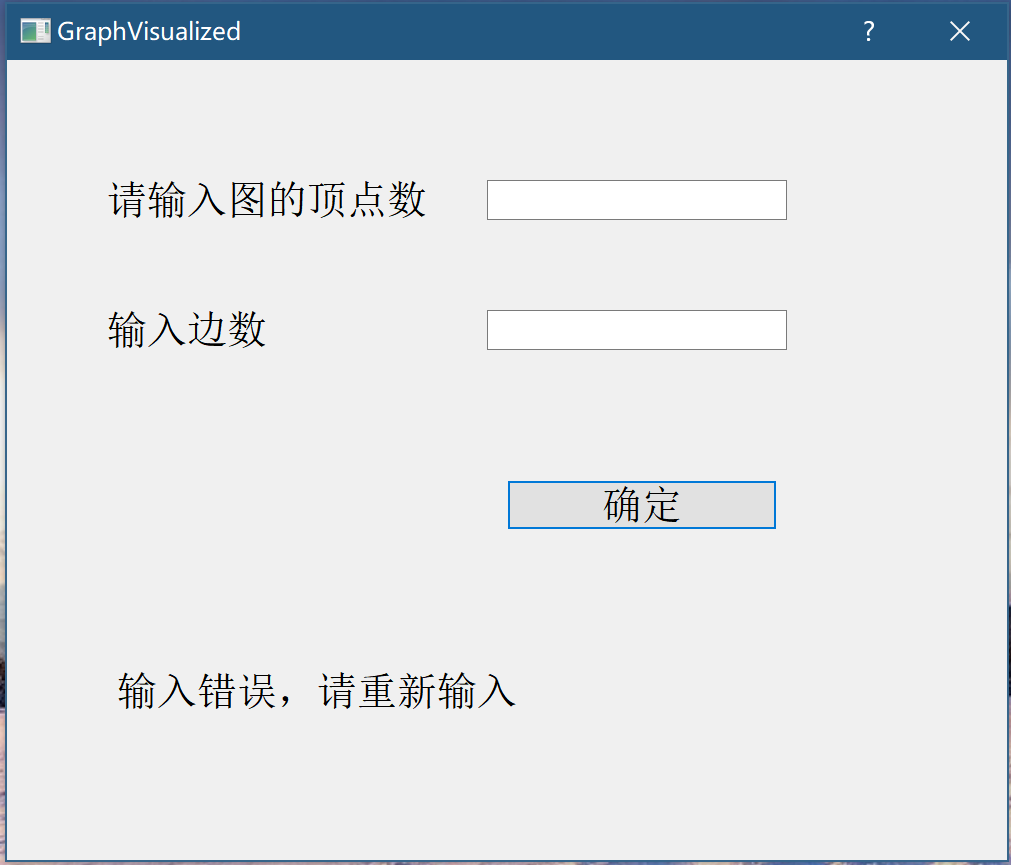
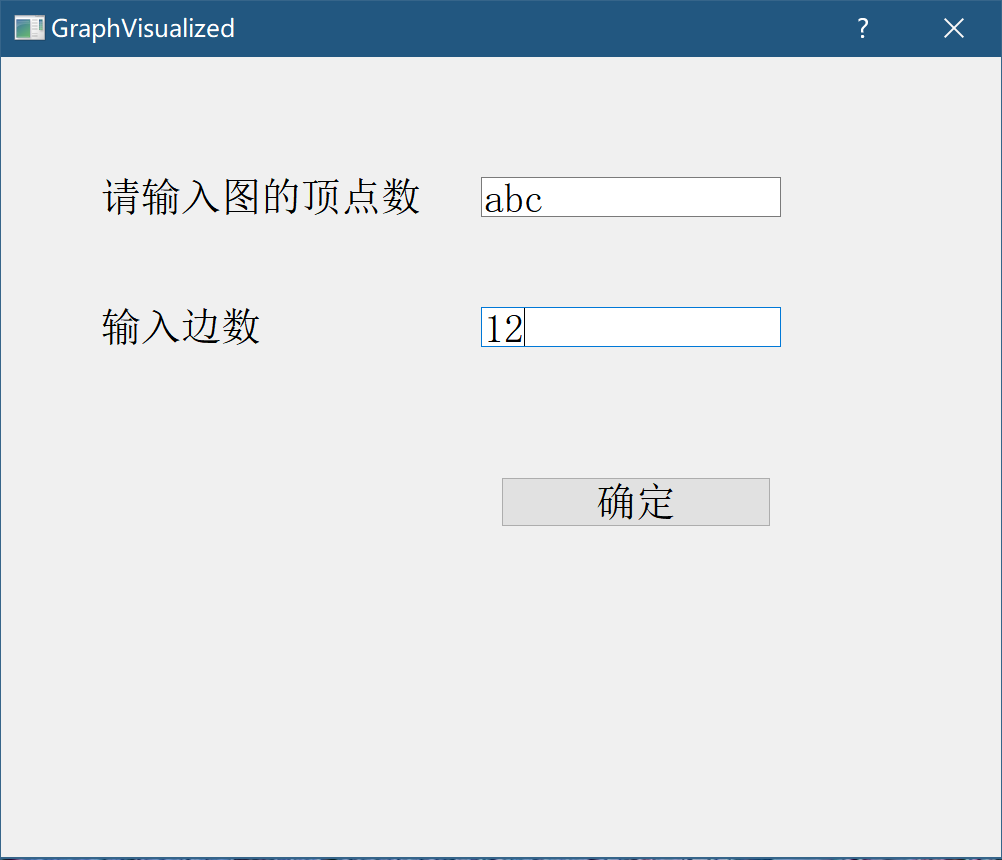


稳定性：

程序可以顺利地执行，不会异常退出

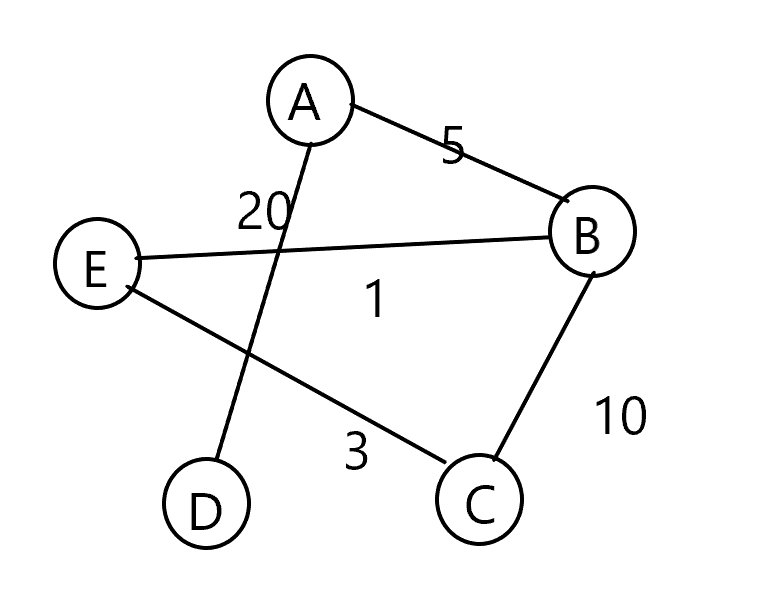
容错性：

在输入错误时，程序可以提示输入错误信息，清空输入，并等待新的输入值

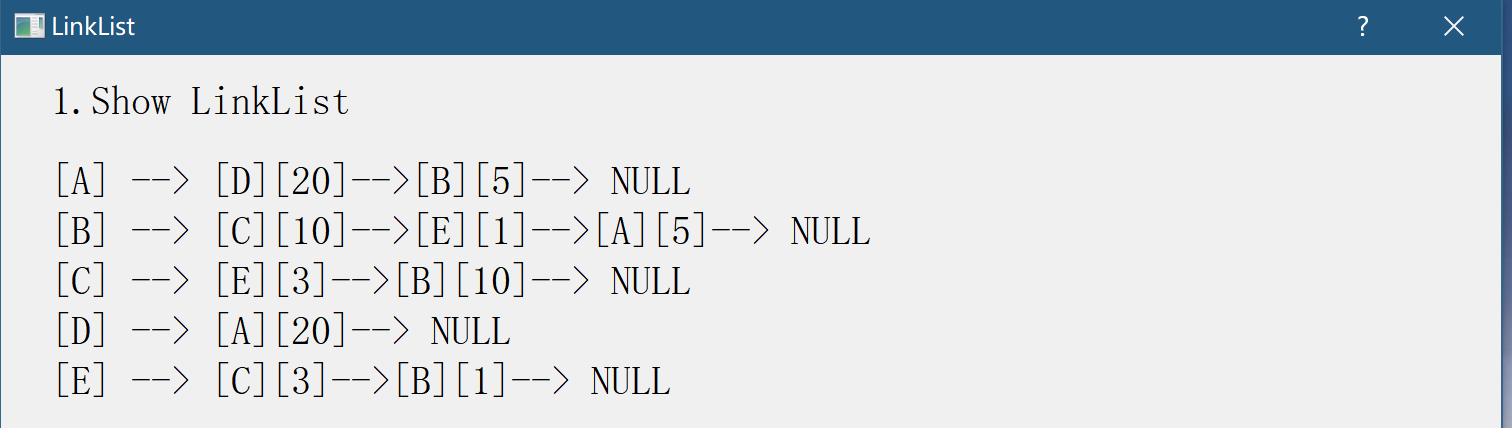


（3）运行案例

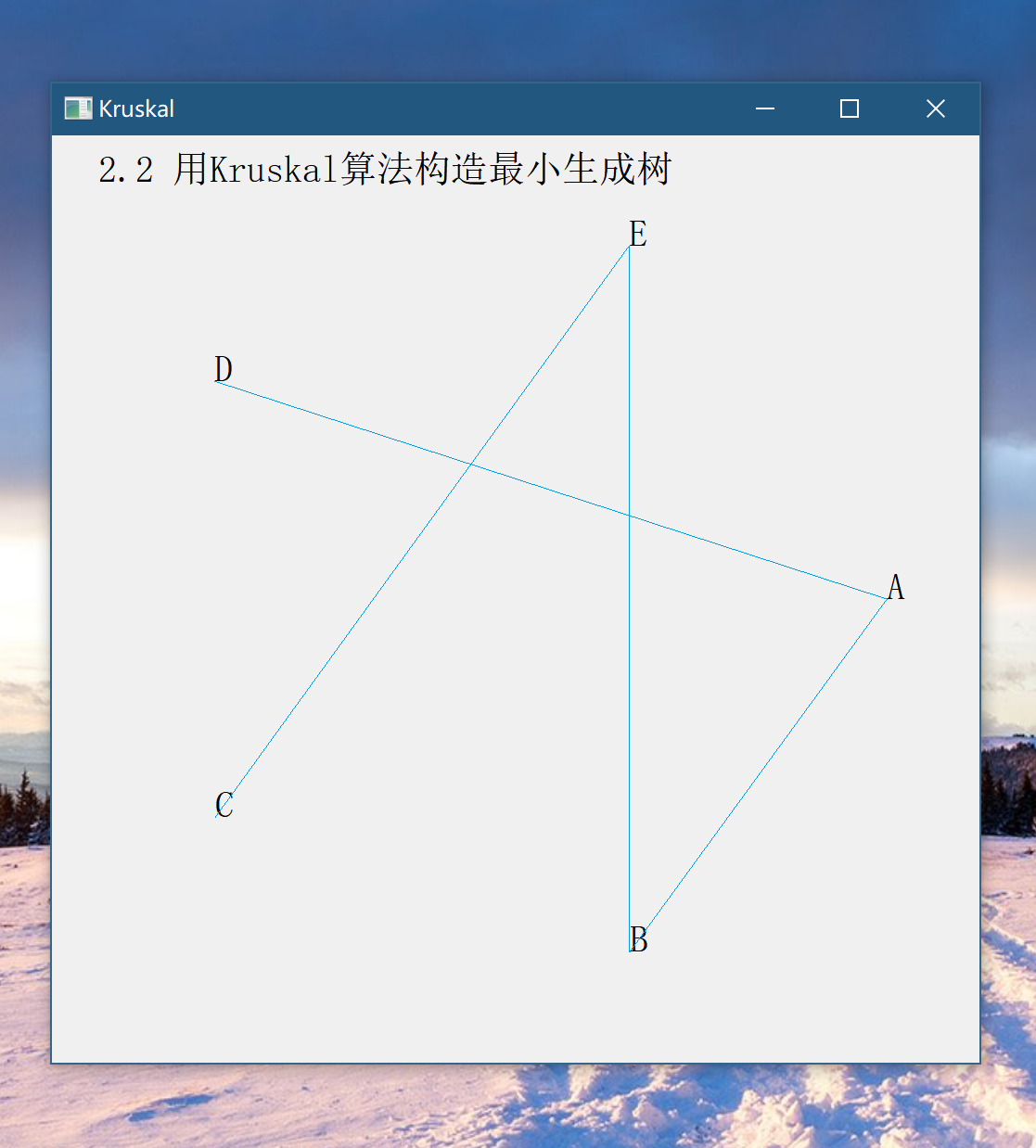
案例1 将如下的无向网输入



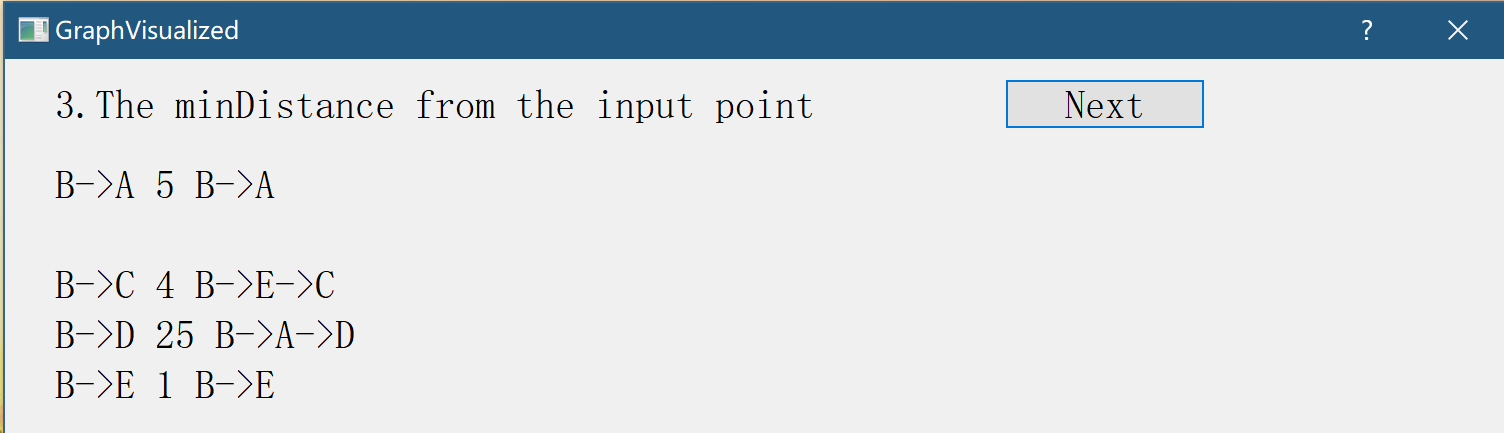
显示邻接链表：



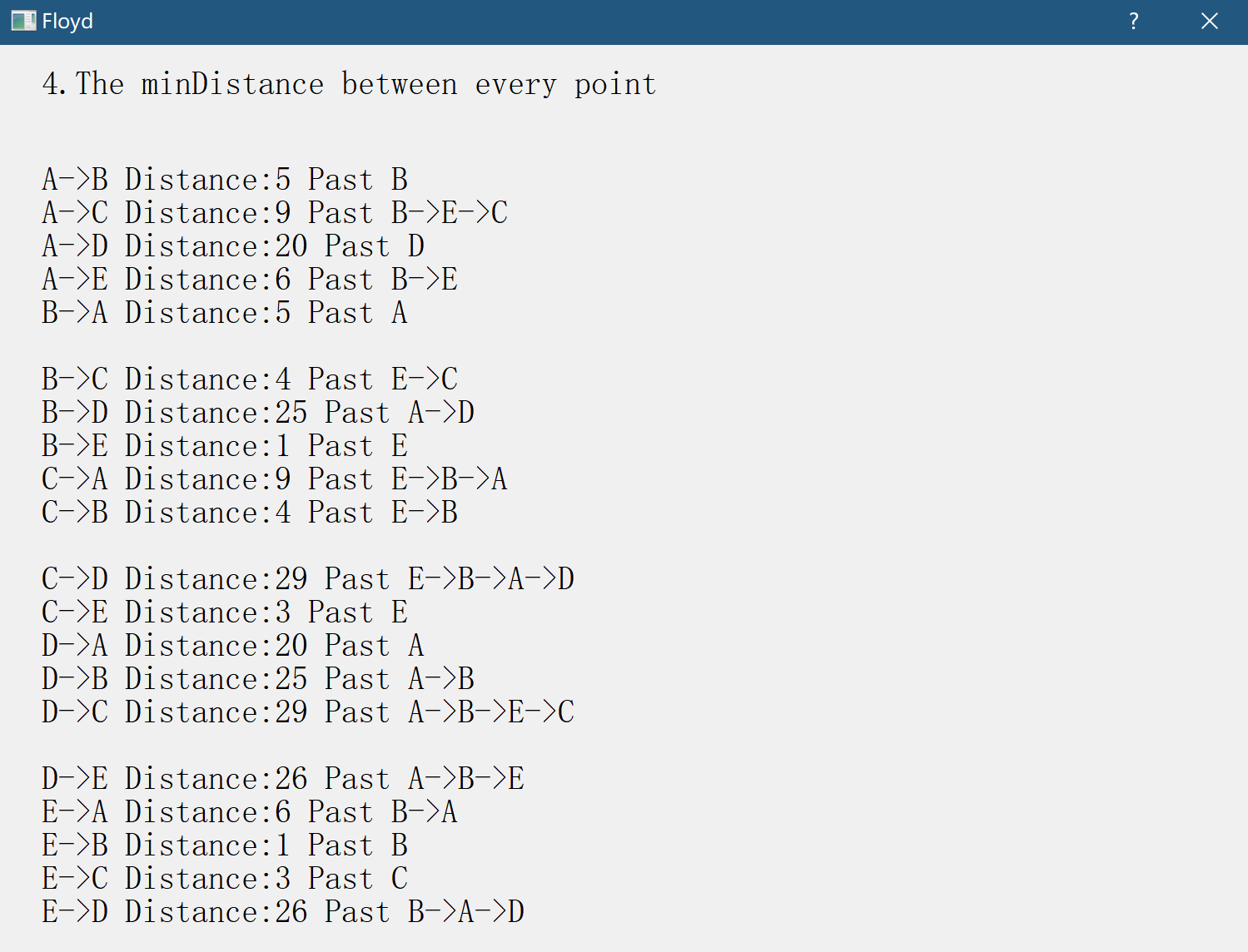
构造最小生成树：



求从B点出发的最短路径：

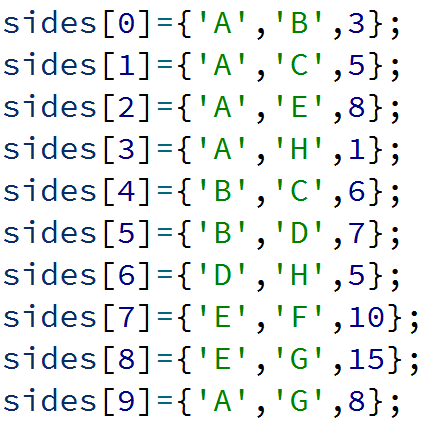


求每对点之间的最短路径：

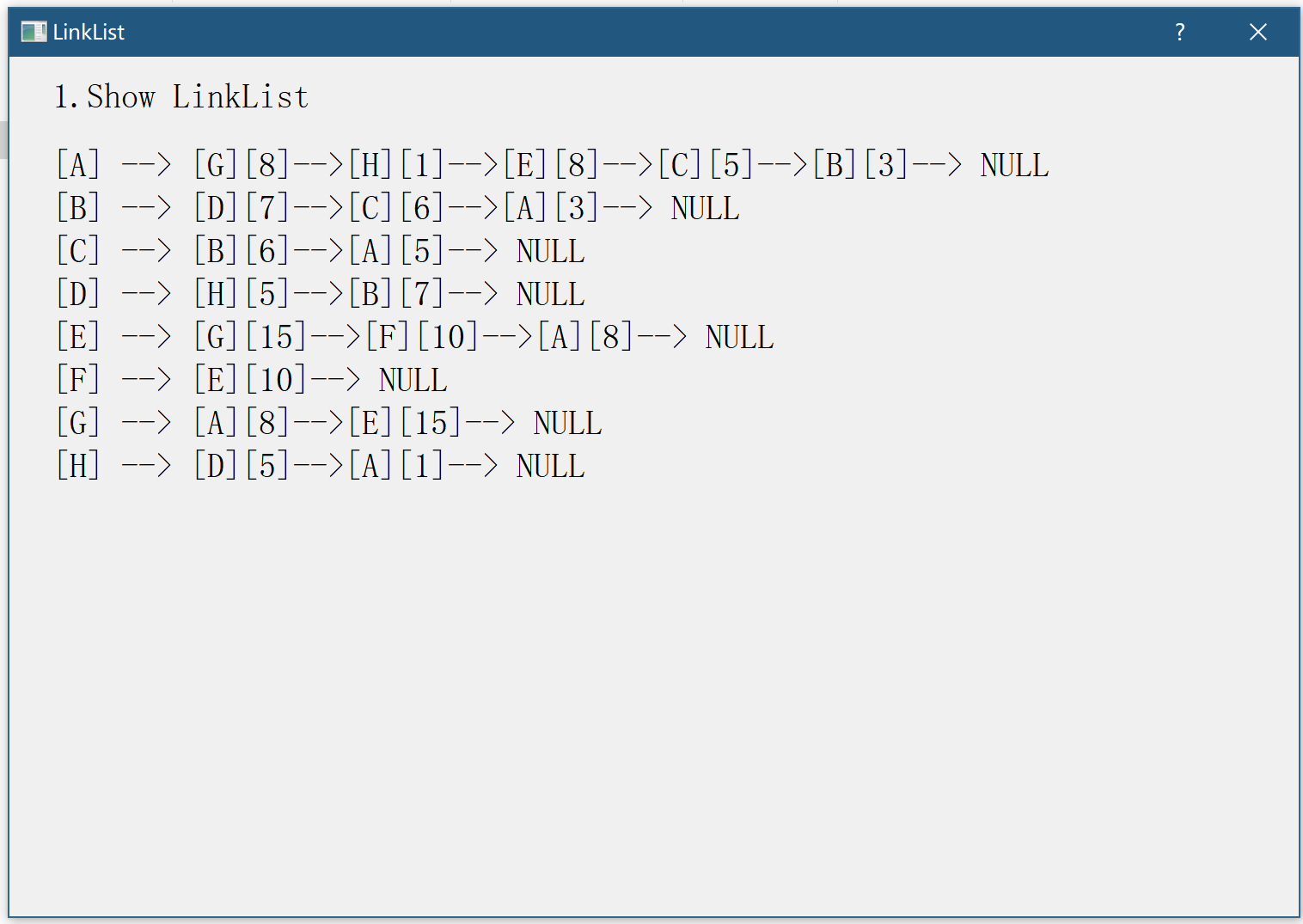


案例2 程序中的默认图

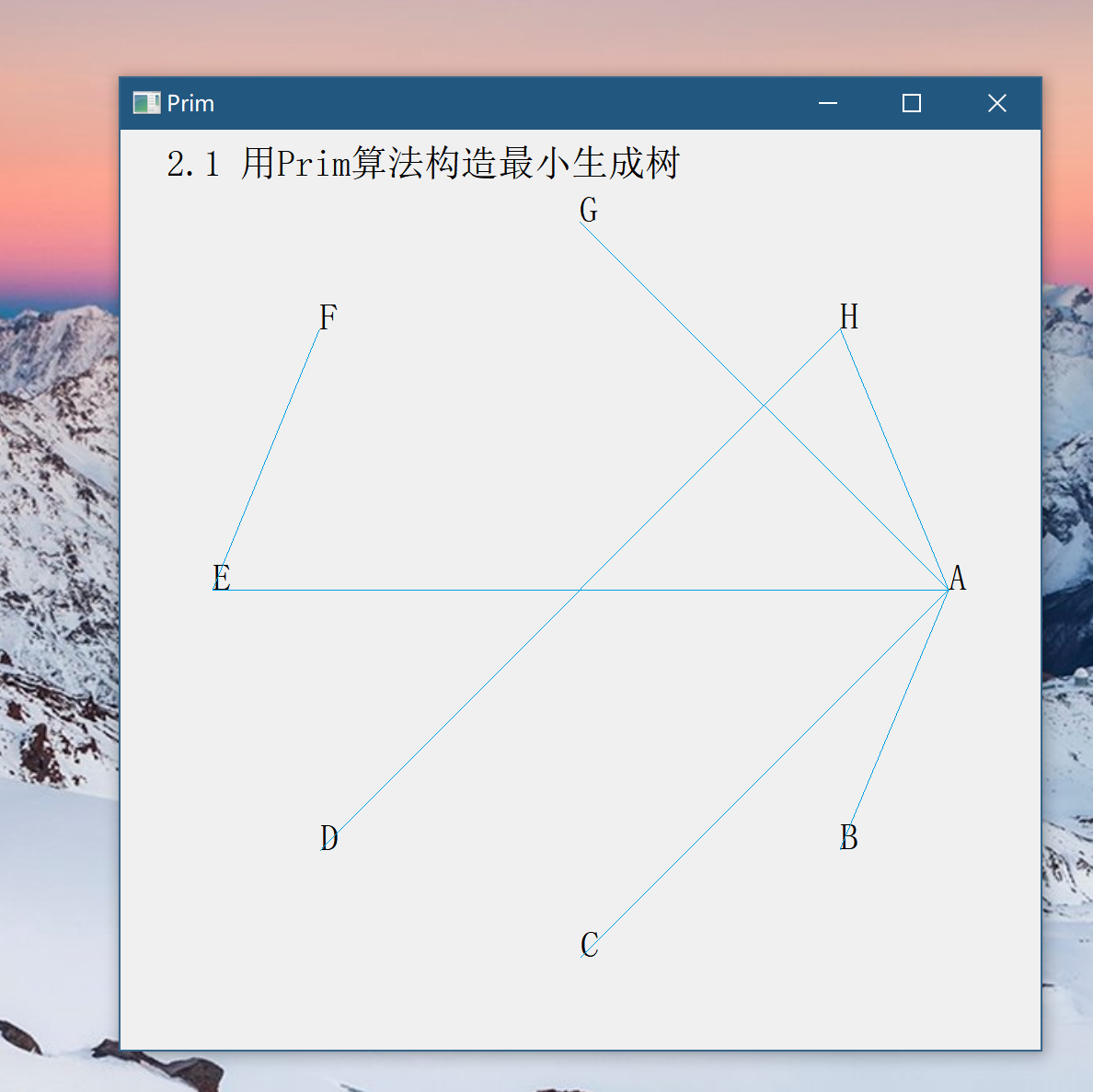
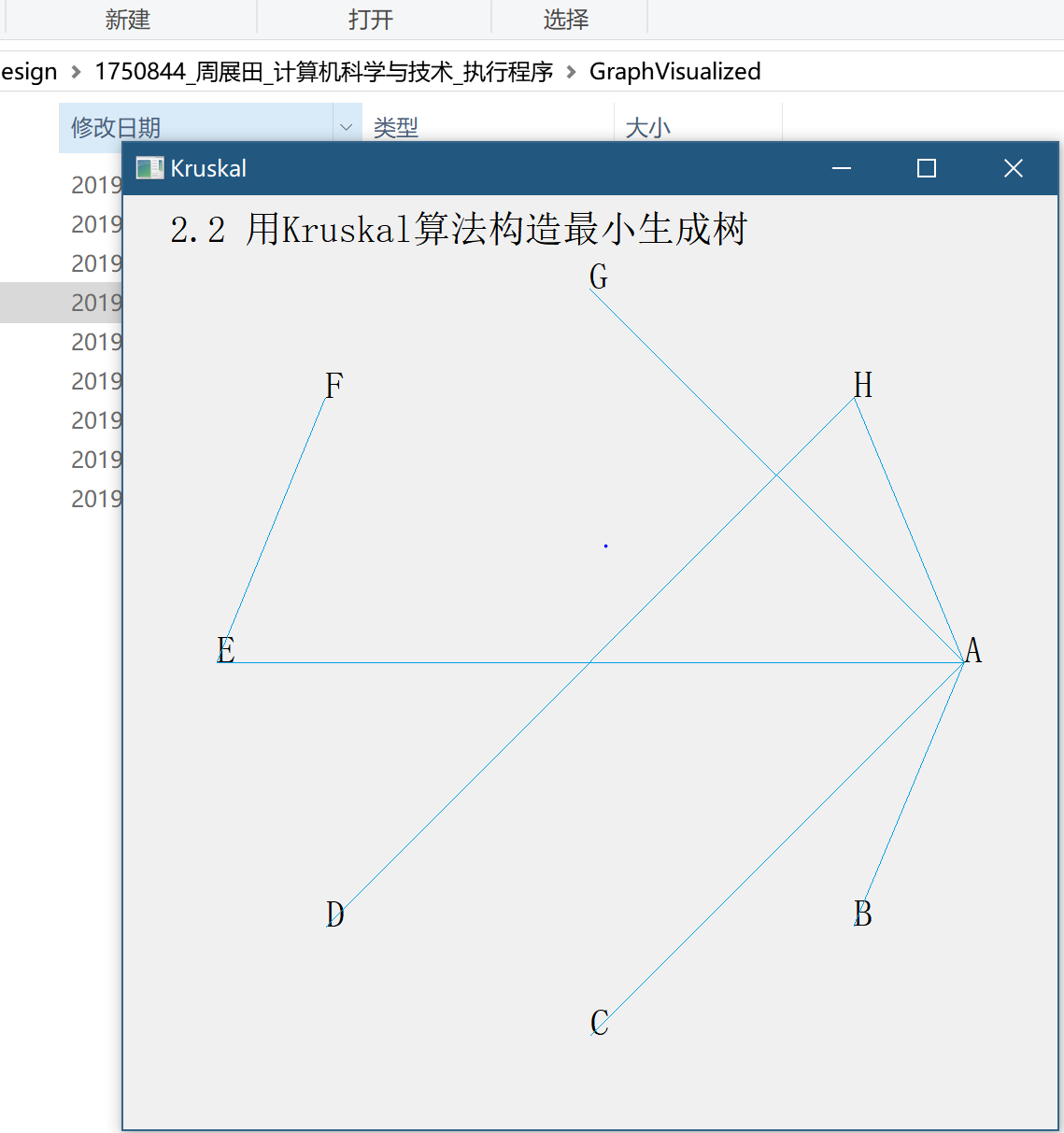
共有8个顶点，10条边，边的结构如下：



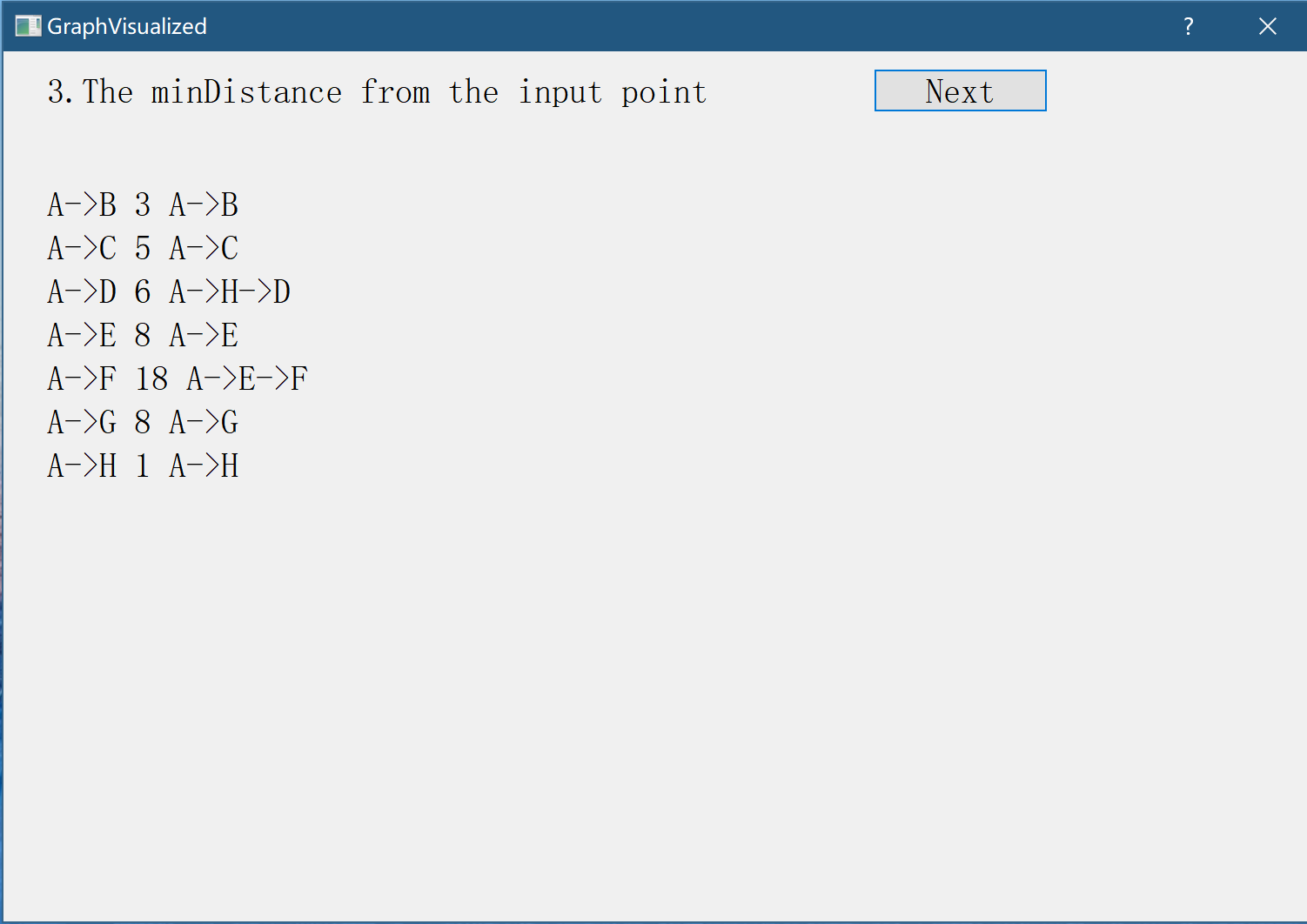
邻接链表：



最小生成树：

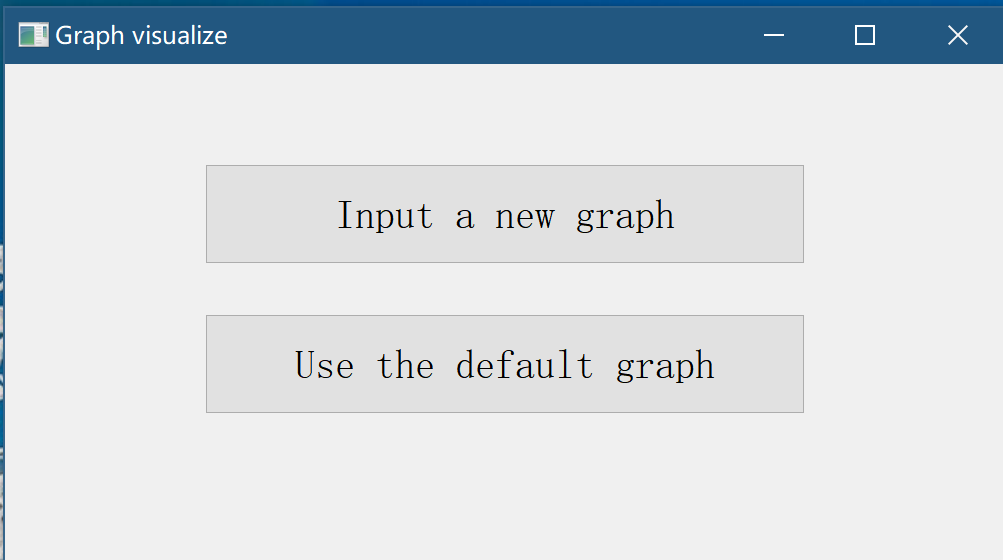
最短路径：



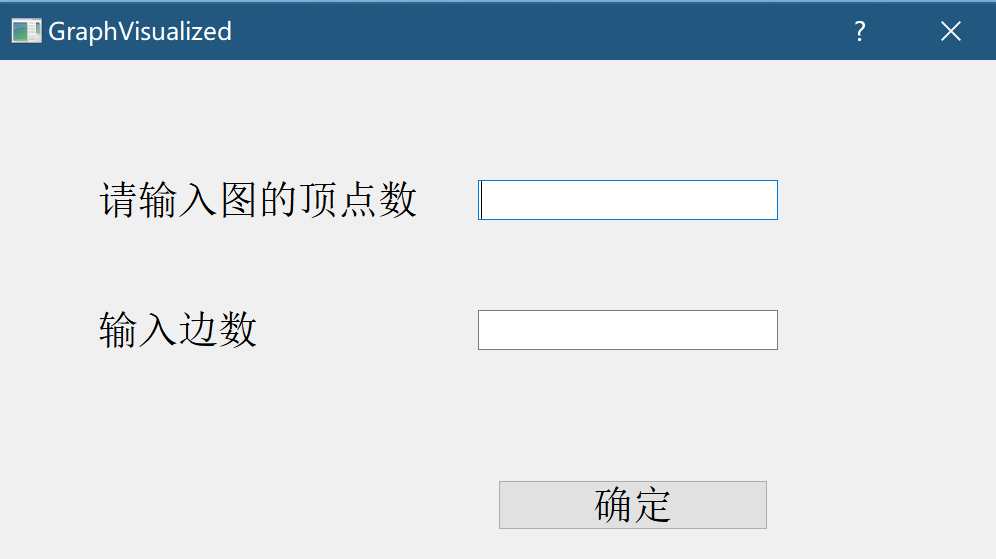
**1.7操作说明**

（1）运行程序，进入选择界面

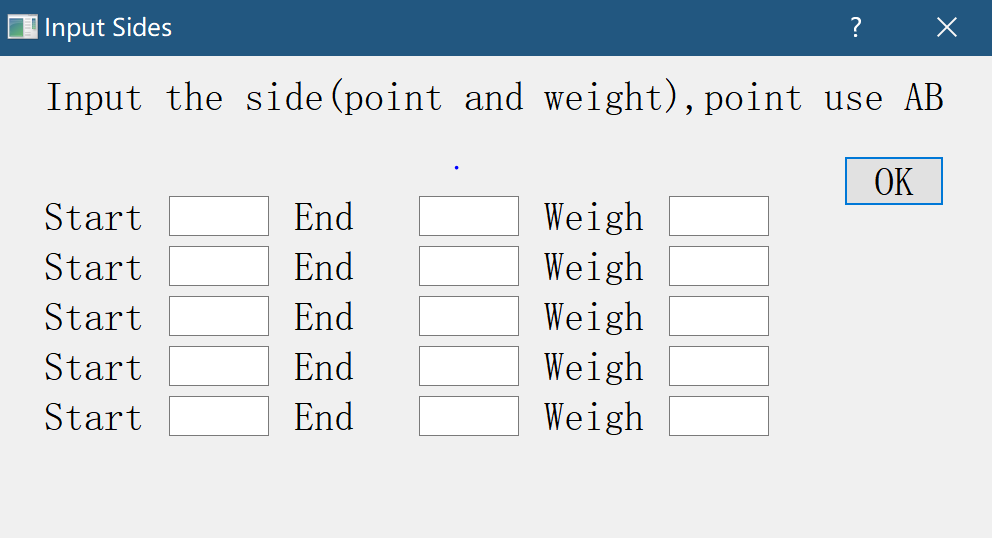
有两个选项，第一个选择项是输入图的数据，第二个是使用程序中预置的图进行演示。



选择“Input a new graph”后进入第一个输入界面，输入图的顶点数和边数。



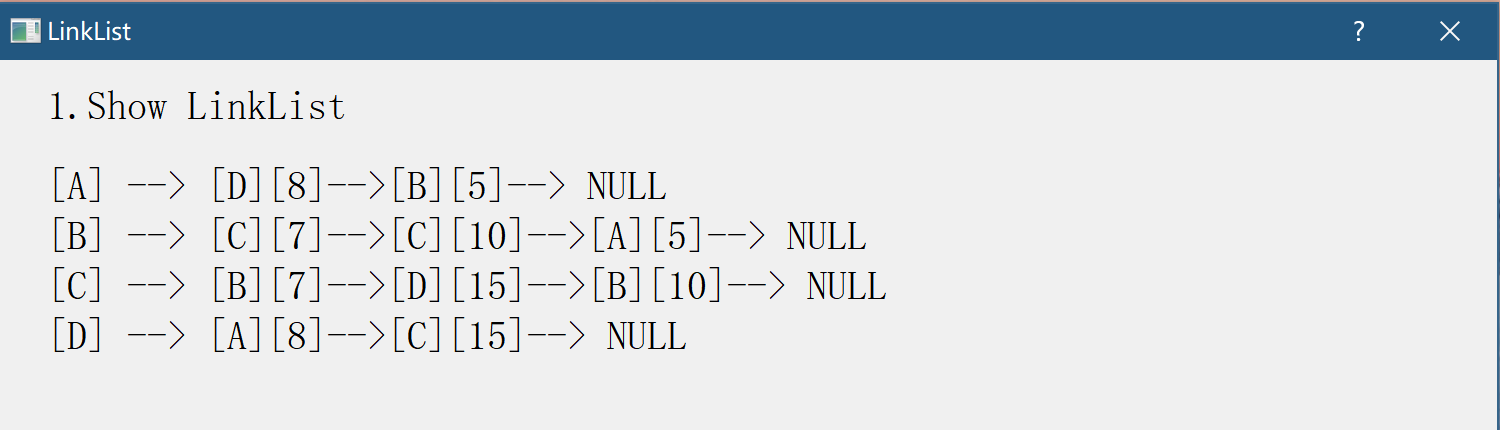
输入顶点和边数并点击确认后，进入第二个输入页面，此时需要输入图的每条边的两个顶点和边的权值，输入框数量的多少会根据之前输入的边数调整。



在输入每条边后，程序会开始创建图，并显示邻接链表

若在第一步选择了“Use default graph”，则会使用程序中预置的图片，不会出现以上输入界面，直接进入邻接链表界面。

（2）显示邻接链表

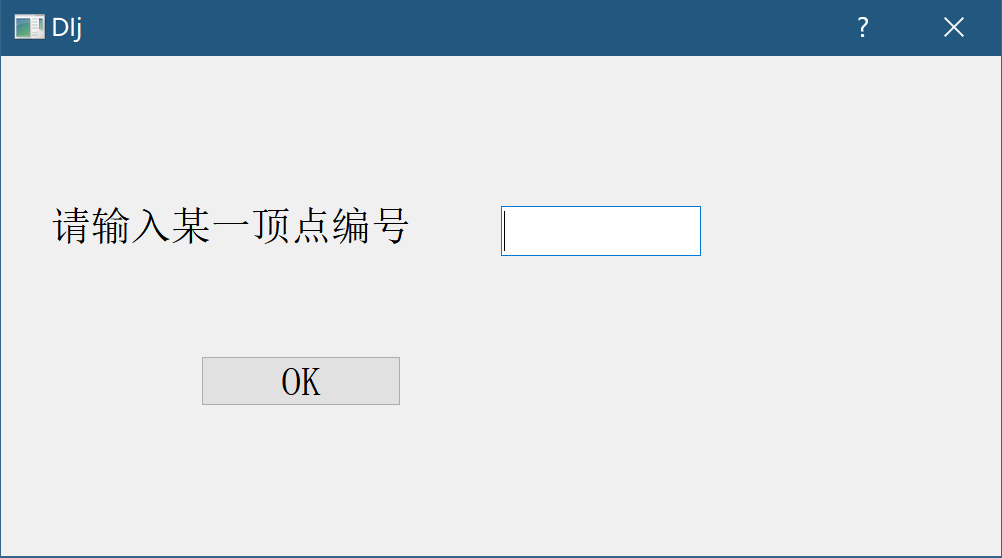


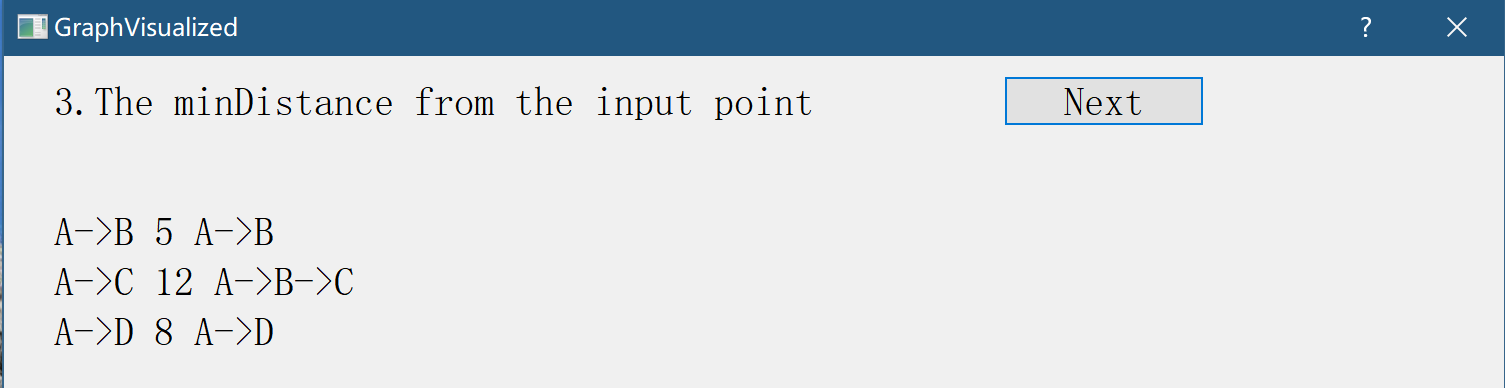
（3）动态构建最小生成树

点击关闭邻接链表界面，程序自动进入两种算法动态生成最小生成树的演示，只需等待，无需操作，演示结束后，会自动进入下一界面

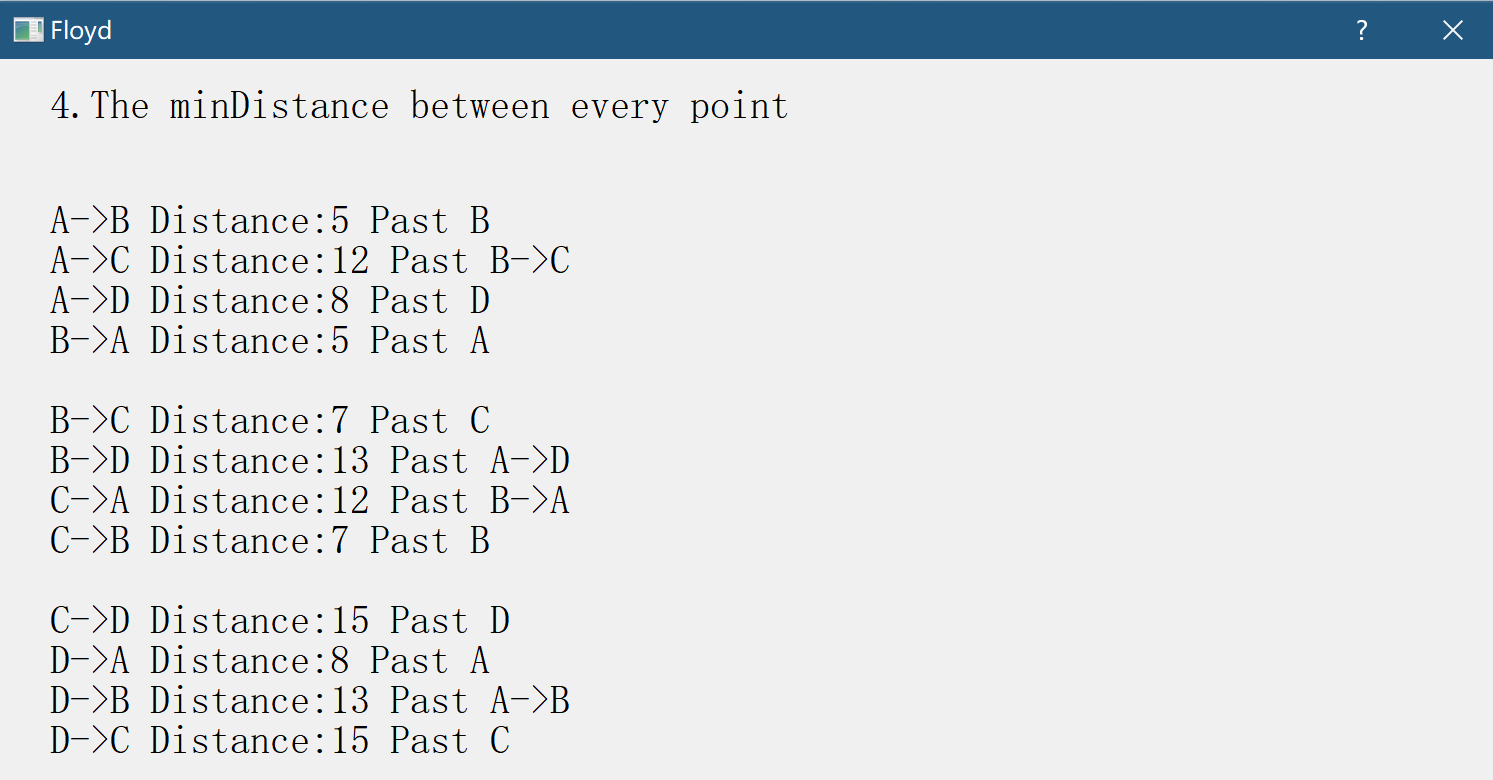
（4）从某一点到其他顶点的最短路径

输入出发顶点的序号，点击OK，程序会显示最短距离情况





（5）点击上图中的”Next”，进入最后一个界面，显示每一对顶点的最短路径



**二、 综合应用设计说明**

**2.1题目**

在产品装配过程中，需根据工艺路径，规划产品每一个零部件的先后装配次序。反之，在产品维修时，需根据该装配次序逆向拆解，直至拆下需要替换的零件，重新按照先前的拆除次序逆向安装。对于汽车，轨道机车、飞机等复杂产品，由于其零部件众多，维修复杂，现需开发一维修教学系统，模拟其拆解维修过程，请设计其中拆解及安装过程仿真算法，实现产品装配次序输入，通过指定需替换的零部件，规划拆解路径和安装路径。

（1）可定义一个产品的装配路径

（2）根据需拆解的零部件，自动规划拆解路径和安装路径。

**2.2软件功能**

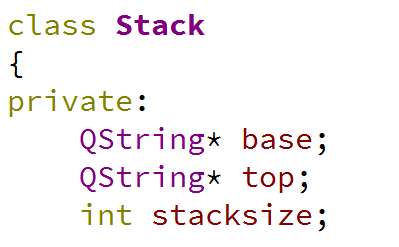
软件可以实现产品装配零件的输入，根据需要更换维修的零件名称，输出更换零件时需要拆卸的零件和重新装配时候的顺序。

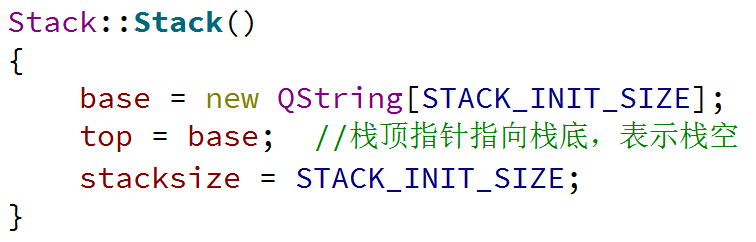
**2.3设计思想**

产品的装配和拆卸过程可以借助栈来完成。在输入原始装配顺序时，将每一个零件Push到栈中，在维修拆卸时，将零件从第一个栈中Pop出来，Push到另外一个栈，维修后重新装配时将第二个栈中的零件重新Pop，Push到第一个栈中。

**2.4逻辑结构与物理结构**

程序使用的逻辑结构为栈，物理结构为顺序结构。





**2.5开发平台**

使用C++语言编程，使用Visual Studio2019，Qt5.13.0 GUI库以及Qt Creator4.9.2开发平台。

软件运行的环境：Windows 10 系统

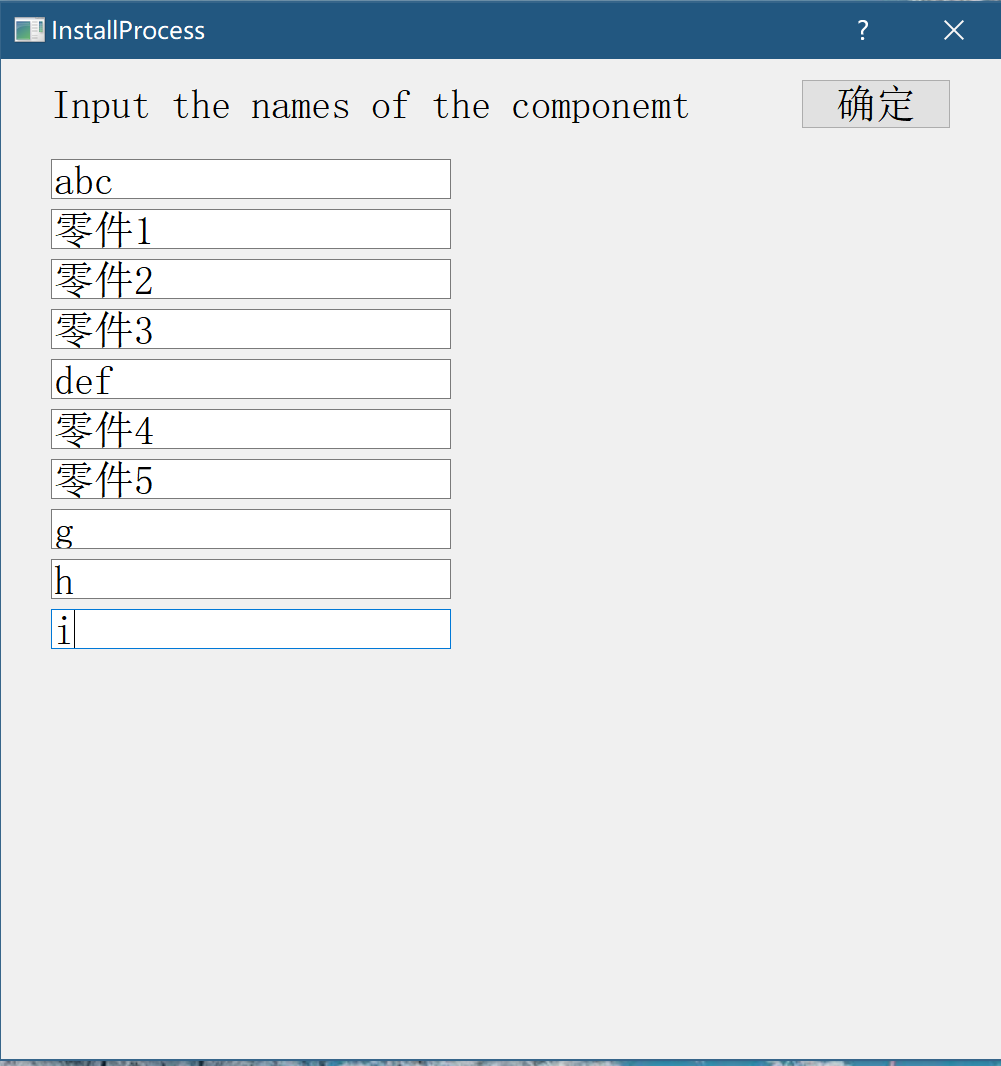
**2.6系统的运行结果与分析**

（1）开发过程

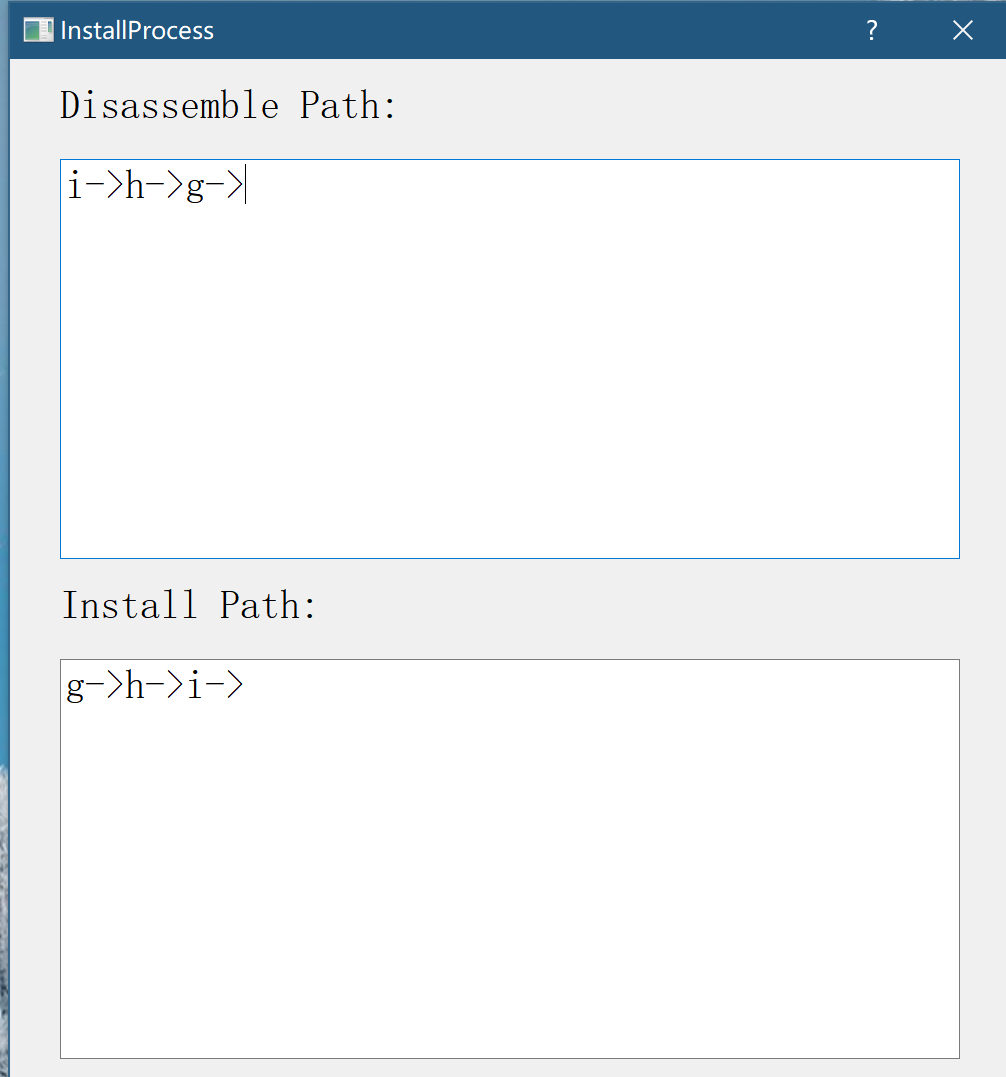
首先用Visual Studio2019编写控制台程序，测试通过后，在Qt Creator中编写GUI程序，使用MSVC7 64bit编译器编译。

（2）达到成果

正确性：程序可以实现装配路径的输入，输入指定零件后完成拆解与装配路径的规划。

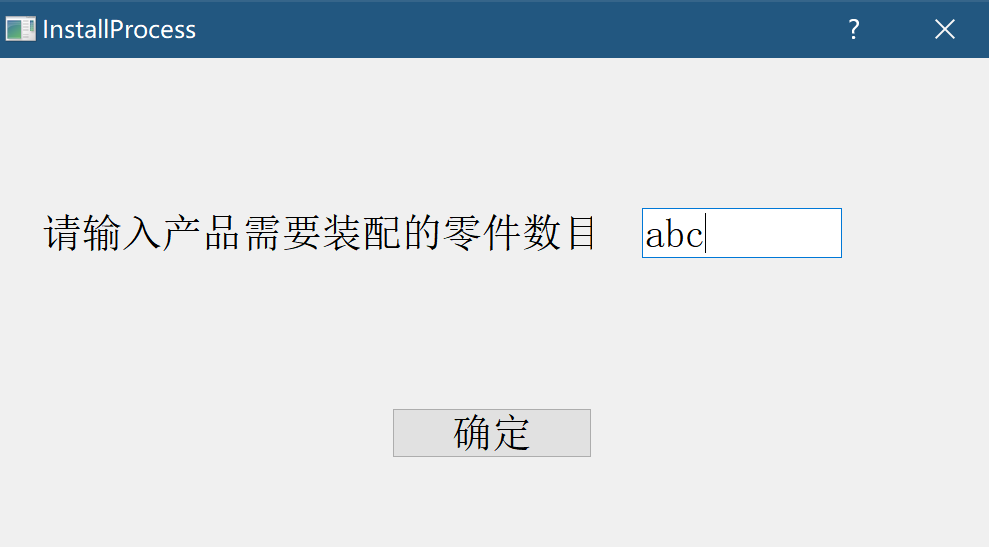


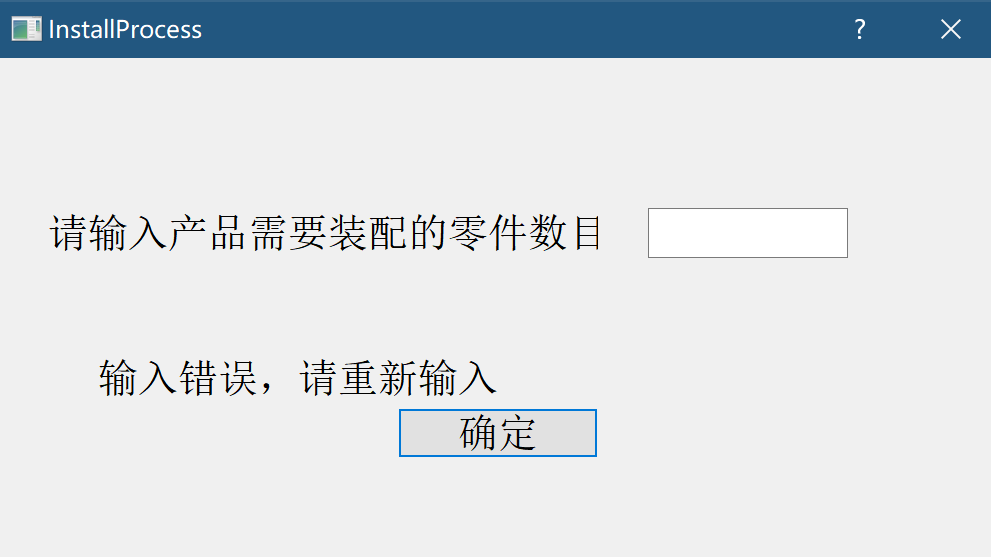
输入维修零件为g



容错性：

对于错误输入，可以给出提示并清空输入



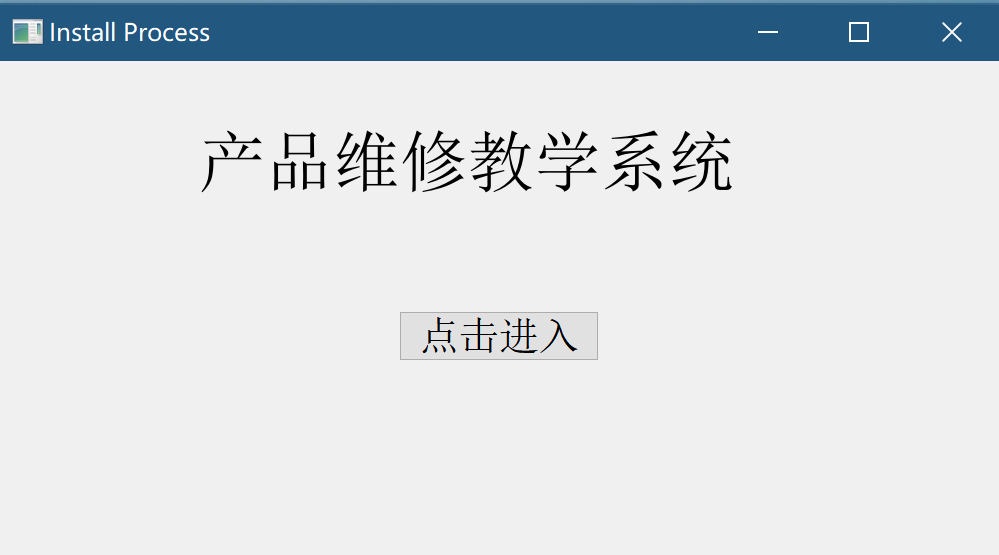


稳定性：程序运行稳定正常，不会异常退出

（3）案例运行 可参考如上案例

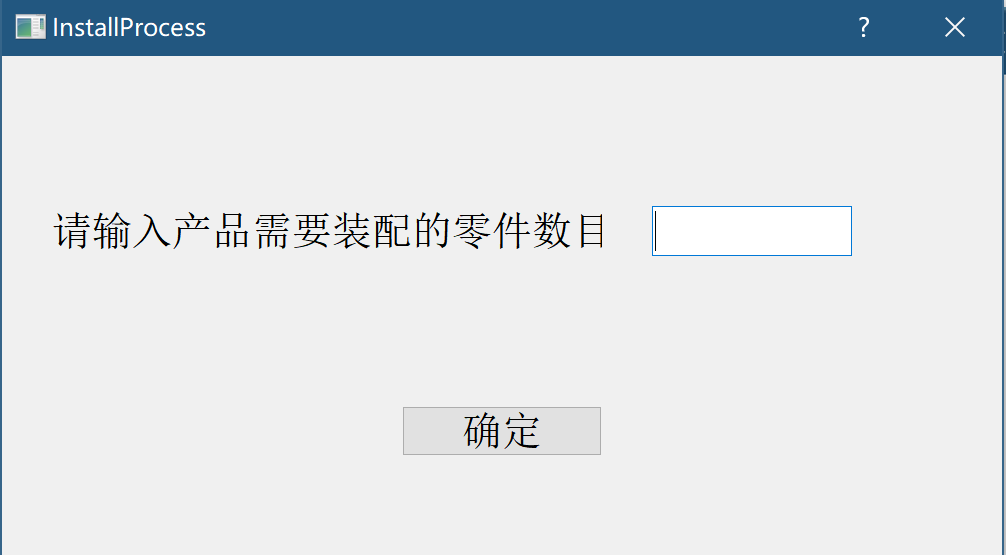
**2.7操作说明**

（1）运行程序，进入开始界面，点击“立即进入”进入程序

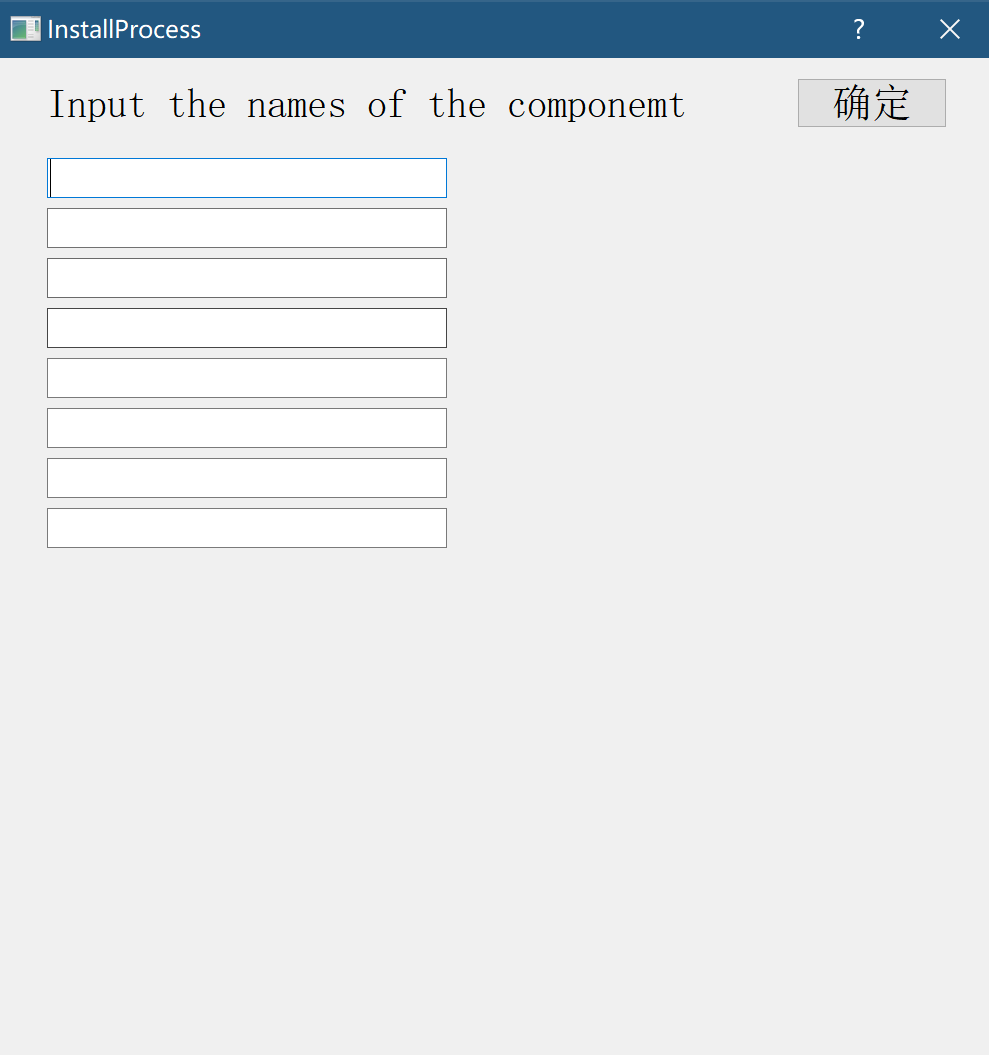


（2）输入界面

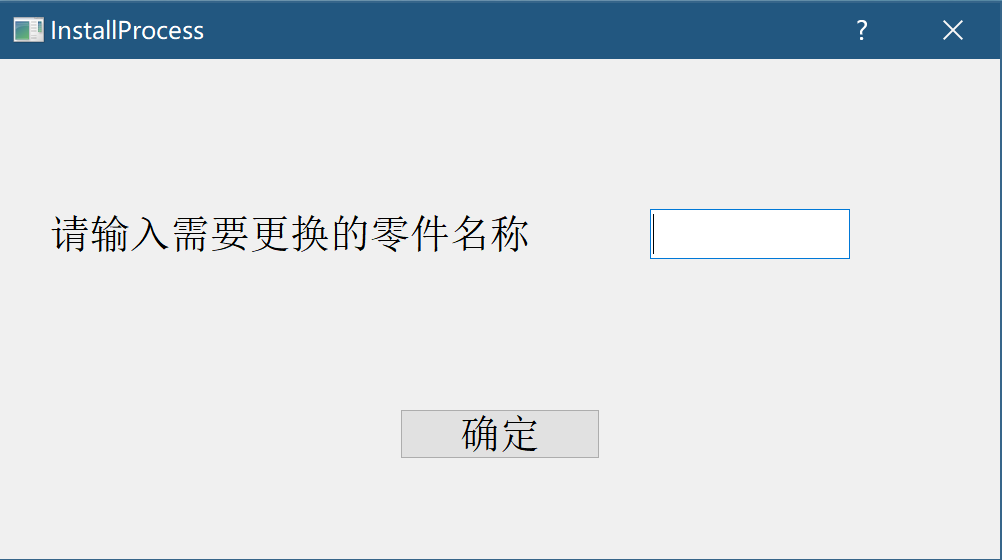
输入产品所需要的零件数目



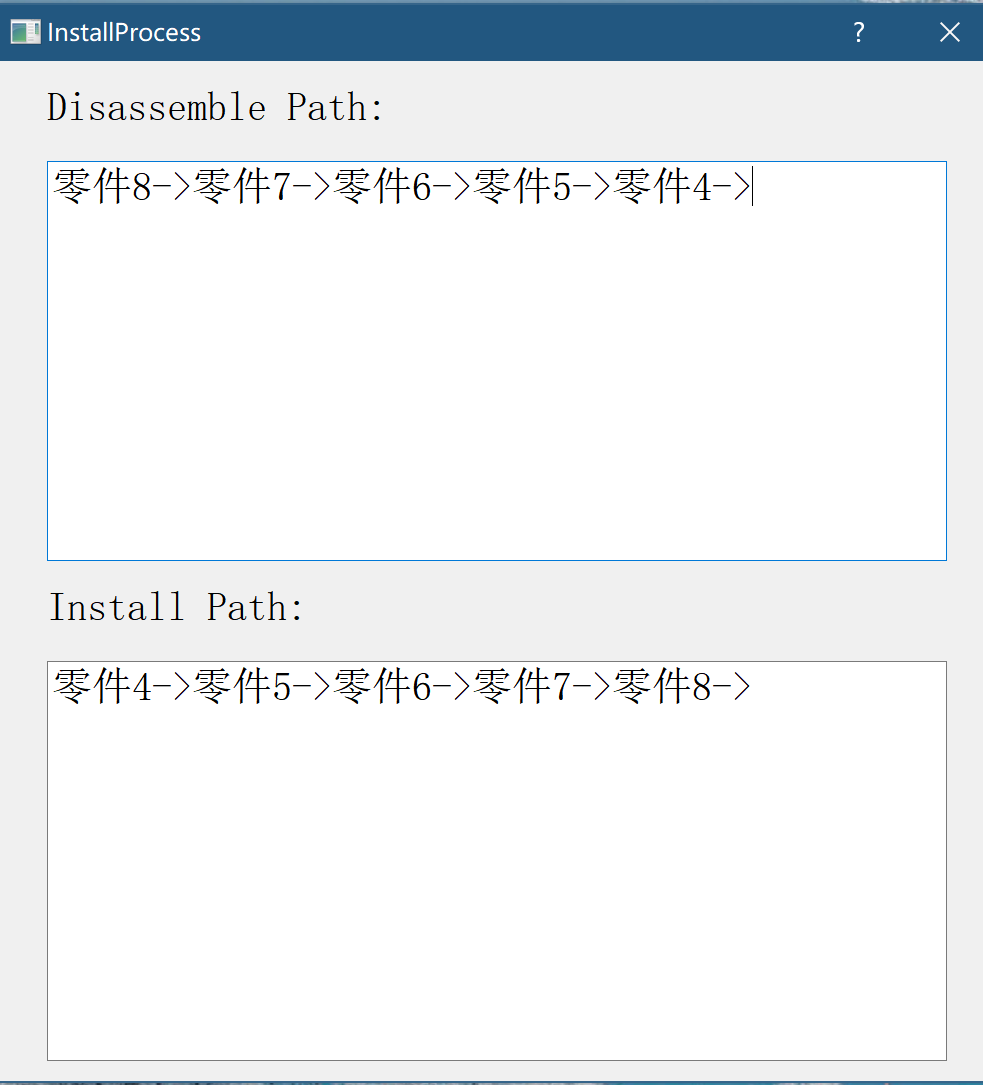
点击确定后，依次输入相应的零件名称，输入完成后点击确定



（3）输入所需要更换的产品零件名称，点击确认



（4）程序会输出拆解与装配路径



**三、实践总结**

**3.1 所做的工作**

前期主要通过网上教程和相关书籍学习Qt编程开发，学习使用Qt的GUI程序设计，由于使用Qt进行开发较为复杂，尤其是Qt中的绘图工具，需要学习的内容较多，所以学习时间较长。中期主要对数据结构理论课程的作业进行一定的修改，使数据结构的算法能使用与此次课程设计。后期进行使用Qt Creator进行软件的功能开发，由于最初设计较乱，同时使用纯代码风格和多个ui进行图形化设计，导致程序编写变得比较困难，所欲在后来对代码进行了重构，使用纯代码的GUI程序设计，完成了两个软件的设计。

**3.2 总结与收获**

复习了图和栈等数据结构，对图的相关操作和算法有了更深的理解，可以较为熟练地完成图与栈地相关算法函数设计，在程序设计时，复习了面向对象编程的知识。

简单学习了使用Qt进行GUI程序开发的方法，对Qt平台下开发可视Windows桌面程序有了基础的认识，了解了Qt特有的信号槽机制，学会了使用信号与槽进行按钮的设计，学会了使用QPainter进行绘图，对paintEvent的原理有了简单的了解，学会了使用定时器实现动态绘制功能，同时对前端设计有了大致的了解。

在使用MSVC2017 64bit进行编译时，由于部分中文字符和全角符号的编码问题报错，使我对与文件的编码方式进行了一些更多的了解。

目前程序只是完成了简单的功能，UI设计还有进一步的改进空间。在完成第一部分后，对Qt的控件及其操作有了更熟练的掌握，在第二部分的软件中，应用更加熟练。

本次课程设计还提高了资料收集能力和个人编程能力。

**四、 参考文献**

[1]严蔚敏 吴伟民.数据结构. 北京：清华大学出版社，2010

[2]陆文周.Qt 5 开发及实例（第2版）. 北京：电子工业出版社，2016

[3] <http://c.biancheng.net/qt/>