组号: 1

图片包含 游戏机, 画

描述已自动生成

上海大学计算机工程与科学学院

**实 验 报 告**

（数据结构2）

学 期：2021-2022年春季

组 长： 李昀哲

学 号： 20123101

指导教师： 朱能军

成绩评定：

二〇二二年4月5日

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **小组信息** | | | | |
| 登记序号 | 姓名 | 学号 | 贡献比 | 签名 |
| 72 | 李昀哲 | 20123101 | 25％ |  |
| 21 | 唐铭锋 | 20121489 | 25％ |  |
| 20 | 刘沛根 | 20121483 | 25％ |  |
| 22 | 李正宇 | 20121517 | 25％ |  |

|  |  |  |
| --- | --- | --- |
| **实验列表** | | |
| 实验一 | （熟悉上机环境、进度安排、评分制度；分组） |  |
| 实验二 | *有向网的邻接矩阵验证及拓展* | P |
| 实验三 | 无向网的邻接表验证和拓展 | P |
| 实验四 | (*实验题目*) |  |
| 实验五 | (*实验题目*) |  |

实验三

1. **实验题目**

无向网的邻接表验证和拓展

二、**实验内容**

模仿有向图的邻接表类模板，完成（带权：非负）无向网 的邻接表类模板的设计与实现。要求实现图的基本运算（如增加删除顶点和边等），并增加如下成员函数：

1. CountDegree(v)，统计顶点 v的度；

2. ConnectedComponent()，求图的连通分量数目；

3. 验证Kruskal和Prim两种最小生成树算法，并设计或实现除此以外的另一种最小生成树算法，如“破圈法”等（仅考虑连通图）；

4. hasUniqueMinTree()，判断无向网是否存在唯一的最小生成树（仅考虑连通图）。

三、**解决方案**

1、算法设计  
 在用邻接表作为存储表示的无向网的类模板中，可分为顶点结点类模板、弧结点类模板、邻接表类模板。在顶点结点类模板中定义了两个数据成员，其一是data域，它存储顶点信息；其二是firstarc域，它是一个指针，指向从该顶点出发的第一条弧结点。在弧结点类模板中定义了三个数据成员，其一是adjVex域，它存储弧头顶点的序号；其二是weight域，它存储这条弧上的权值；其三是nextarc域，它是一个指针，指向从该顶点出发的下一条弧结点。在无向网邻接表类模板的定义中，顶点表vexTable是一个顶点数组，每个顶点对应一个数组元素，用以存放顶点信息和指向从该顶点出发的第一个弧结点；还定义了vexNum、vexMaxNum和arcNum三个数据成员，分别记录图中当前顶点数目、允许的顶点最大数目和弧数；为了在图的遍历等算法中记录顶点是否访问，在此定义了一个标志数组tag需要时用以记录顶点的访问状况,另外数据成员infinity表示无穷大的值。对于插入边的操作，无向图和有向图略有不同，无向图需要考虑这条边关联的两个顶点，在两个顶点的邻接表中都需要加入它们的信息。

1. 统计顶点v的度:创建一个指针，从顶点v出发，依次遍历其所连接的弧结点，直到指向null为止，此时到达边链表尾，其间每次遍历用计数器计数，最终返回的计数器值即为顶点v的度。
2. 求图的连通分量数目:用DFS算法对图进行遍历，其中调用DFS函数的次数即为该图的连通分量数目。
3. 验证Kruskal和Prim两种最小生成树算法，并设计或实现了“破圈法”求最小生成树（仅考虑连通图）：

首先我们所学到的Kruskal和Prim算法分别对同一个无向连通图求其最小生成树，并得到了相同的正确结果，从而验证了这两种算法的可行性。

接着，我们开始考虑除这两种算法之外的另一种算法，即“破圈法”，其大致思路是先找到图中所有存在的圈，接着在所有构成圈的边中选择一条权值最大的边并将其从图中删除，然后再次寻找图中是否还存在圈。如有则再次进行选权值最大的边删除的操作，如此重复，直到图中没有圈为止，此时该图即为最小生成树。

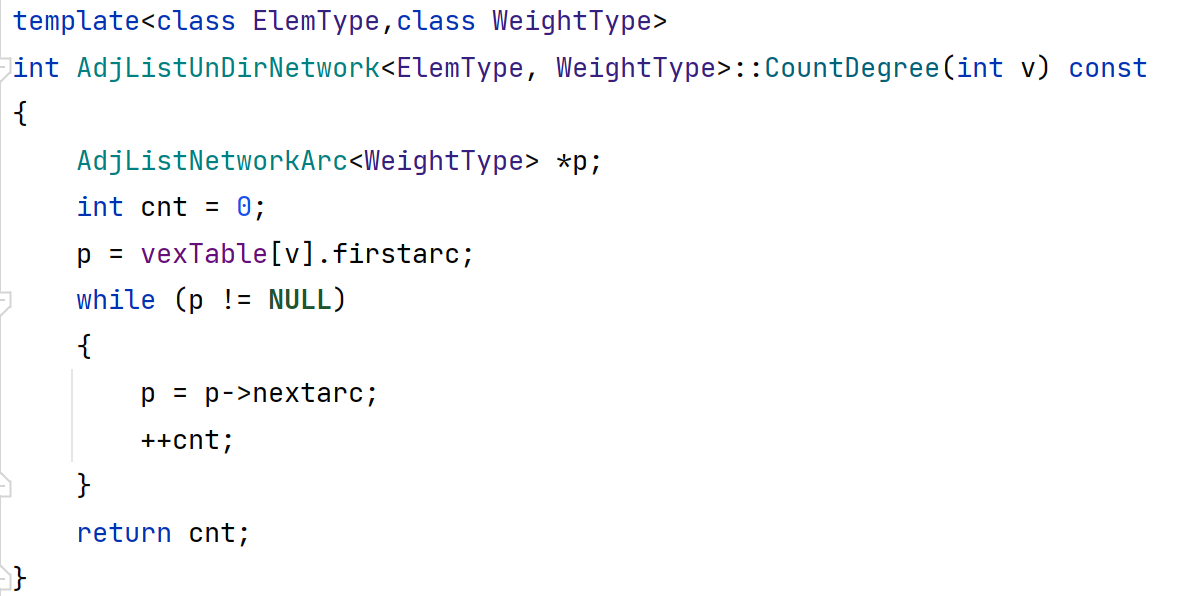
“破圈法”其实也是一种贪心算法，具体代码实现可简述为：

1. 用拓扑分类算法，依次找到图中度为1的顶点，可以保存在队列里，接着遍历队列，依次将各个顶点出队，然后与其邻接的顶点的入度减1，并判断这些顶点在度数减1后是否度数为1，若为则入队，这样往复操作，直到图中已不存在度数为1的顶点，即所有的顶点的度数都大于等于2，那么剩下的边就都在环里了。如果没剩下边，说明没有环，算法结束。
2. 在步骤1结束后，剩下的边就都是环中的边了，找一个权最大的删除。
3. 再进行1操作，直到图中无圈，即所有的圈都已破掉，剩下的就是最小生成树了。
4. 判断无向网是否存在唯一的最小生成树（仅考虑连通图）：

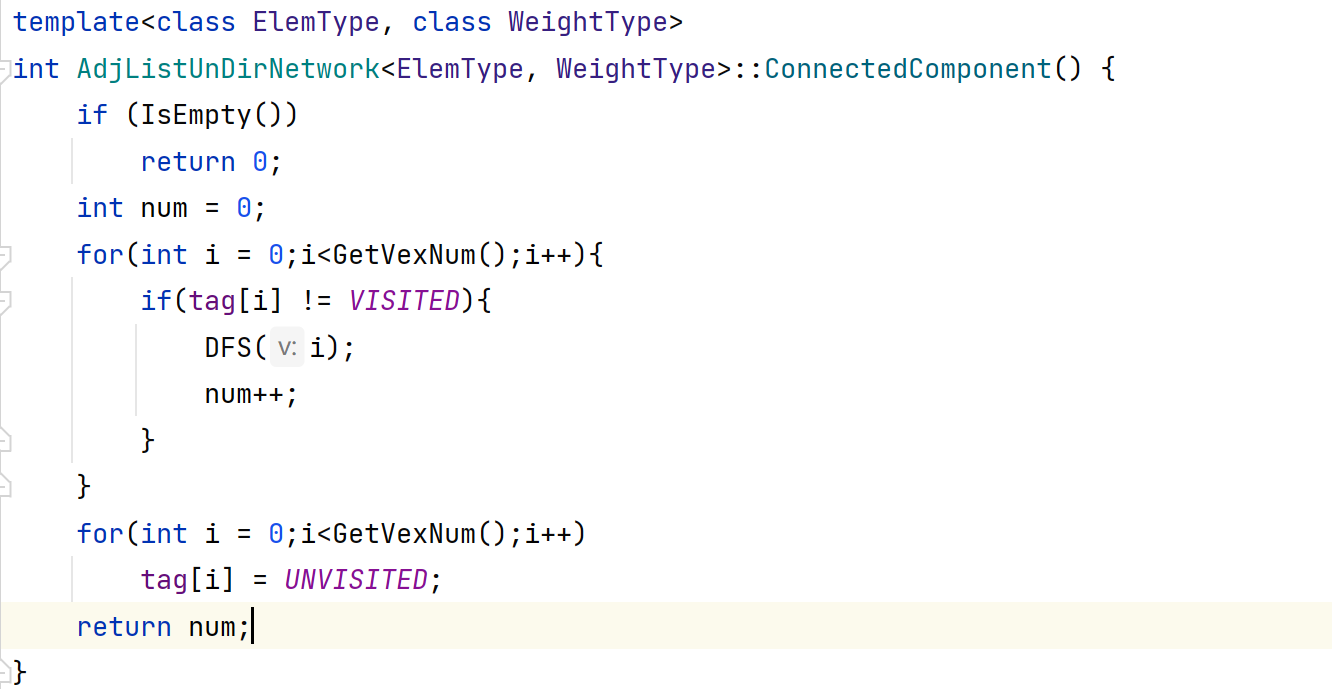
思路可简述为：使用Prim算法求最小生成树的过程中，若有两个顶点到某一顶点的边的权值相同且是最小权值，或者若有一个顶点到另外两个顶点的边权值相同且为最小值，则最小生成树不唯一，否则存在唯一最小生成树。

1. 源程序代码

（1）统计顶点v的度



1. 求图的连通分量数目



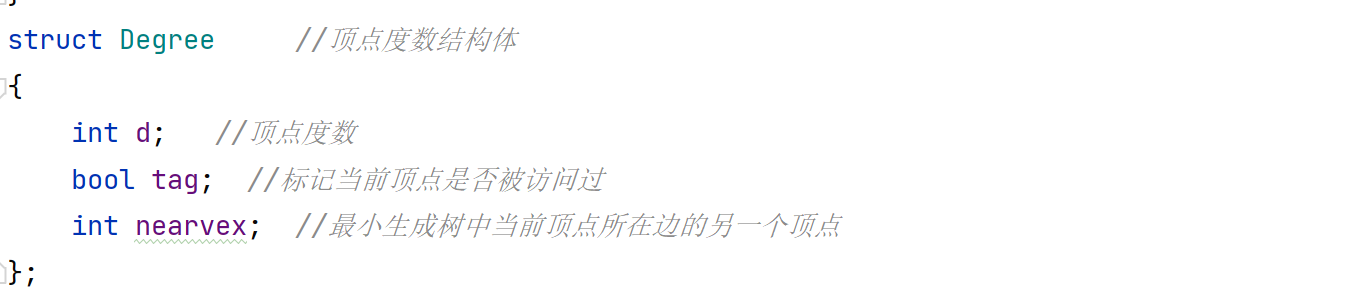
1. “破圈法”求最小生成树

1、**带权边类的定义：**





**2、定义顶点度数结构体**

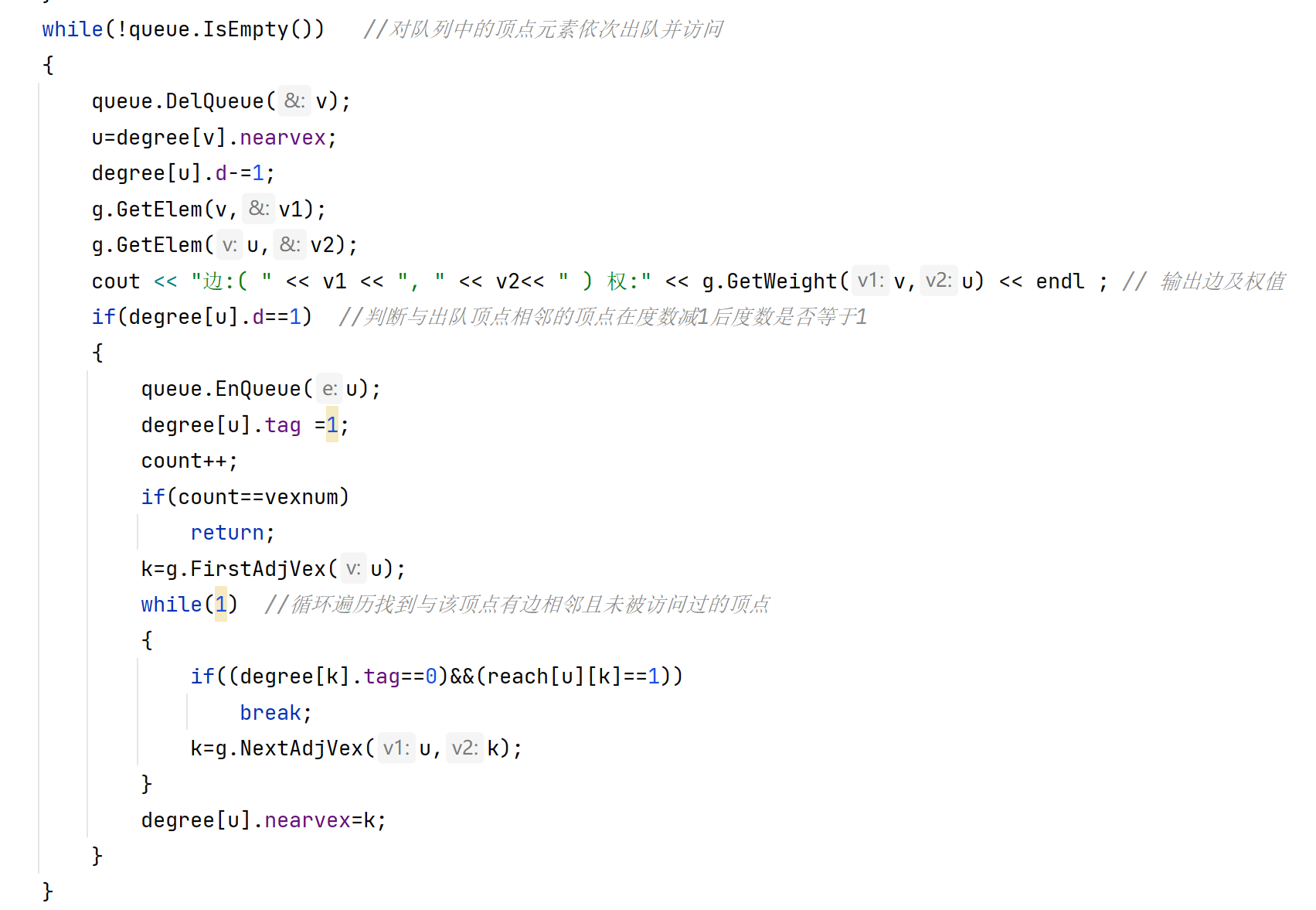


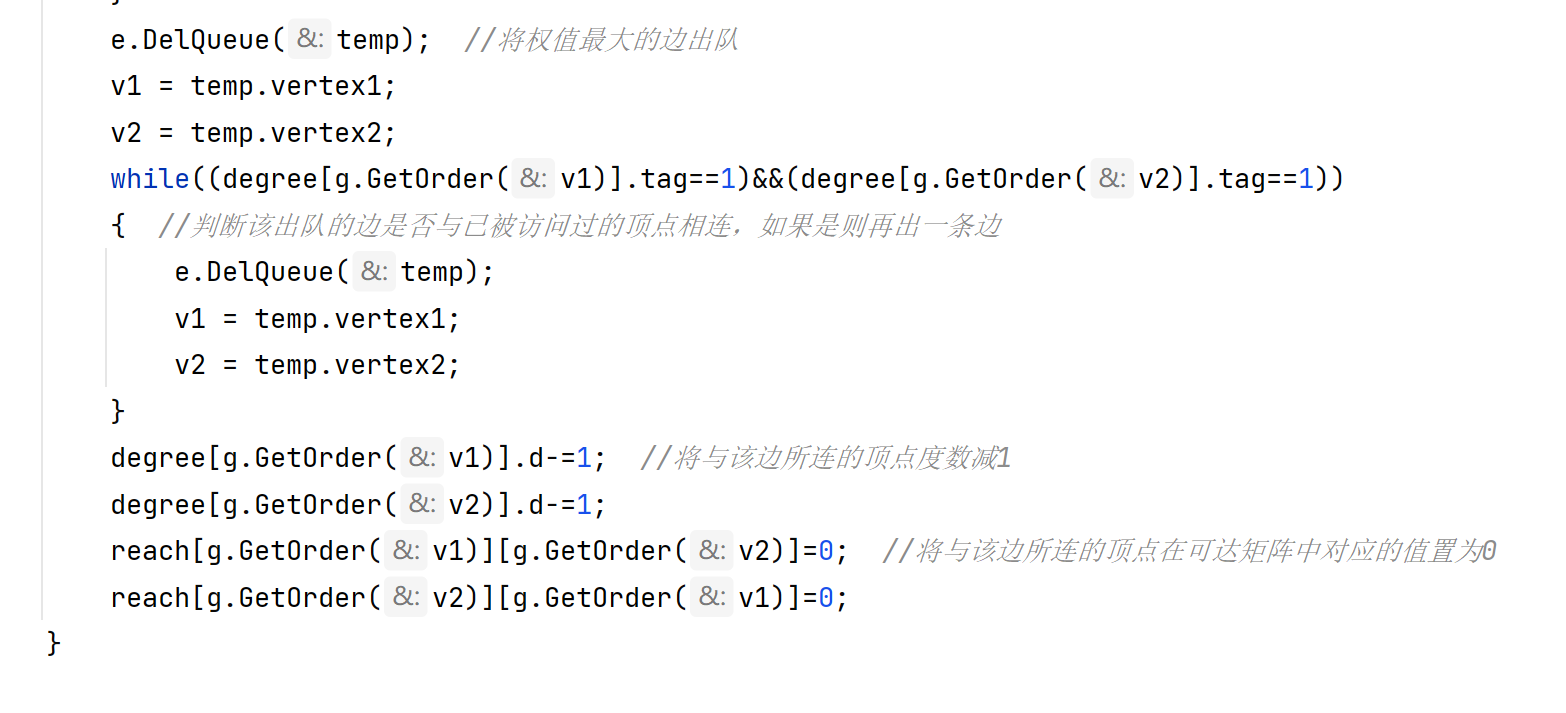
**3、破圈算法具体函数实现**





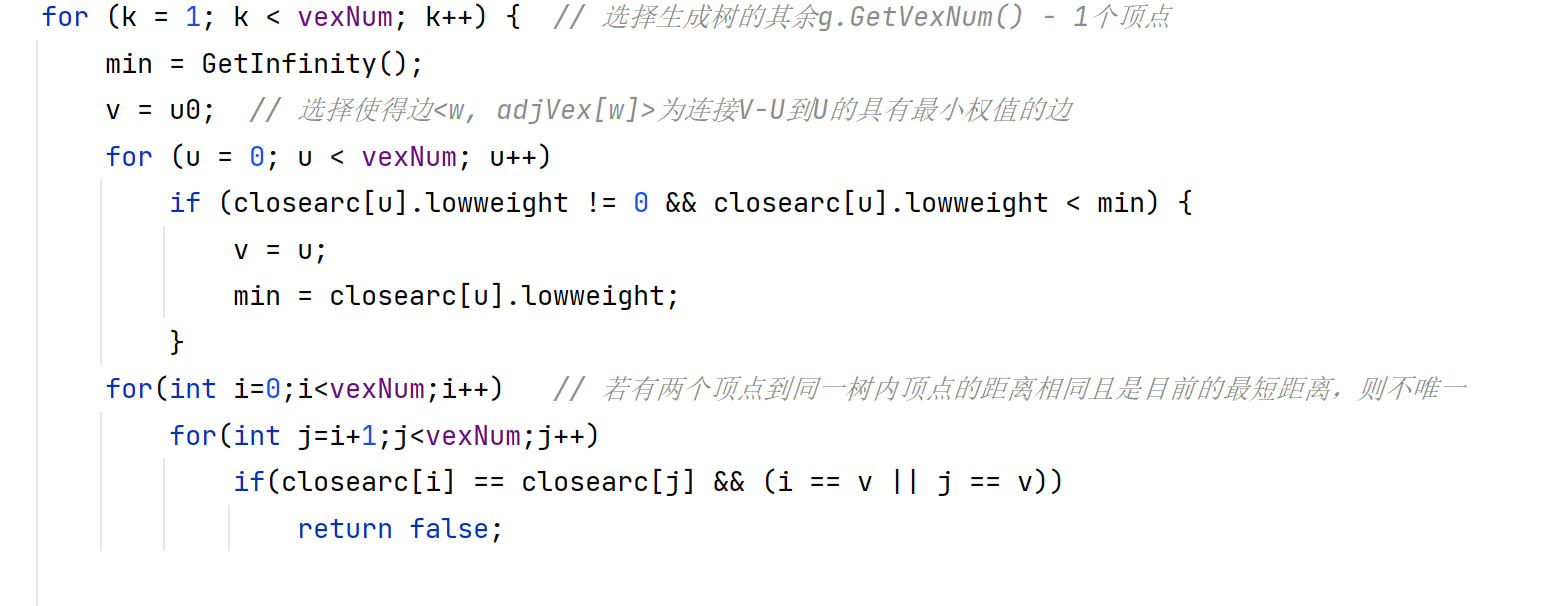


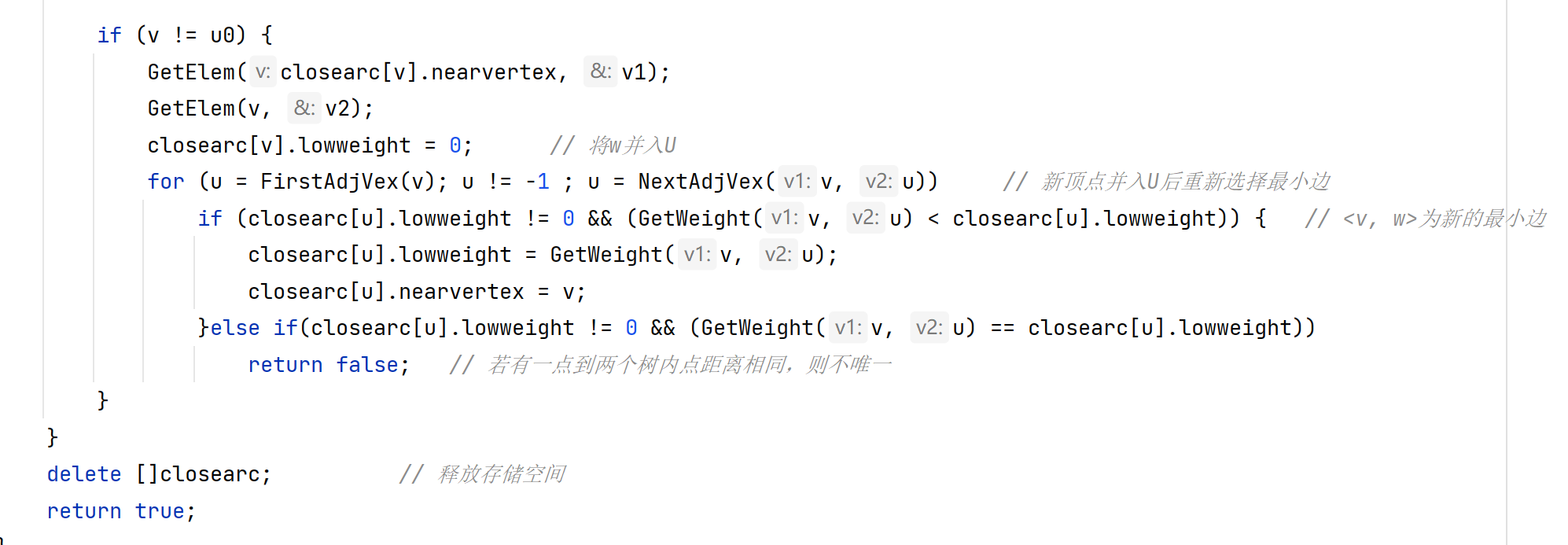




1. 判断无向网是否存在唯一的最小生成树（仅考虑连通图）

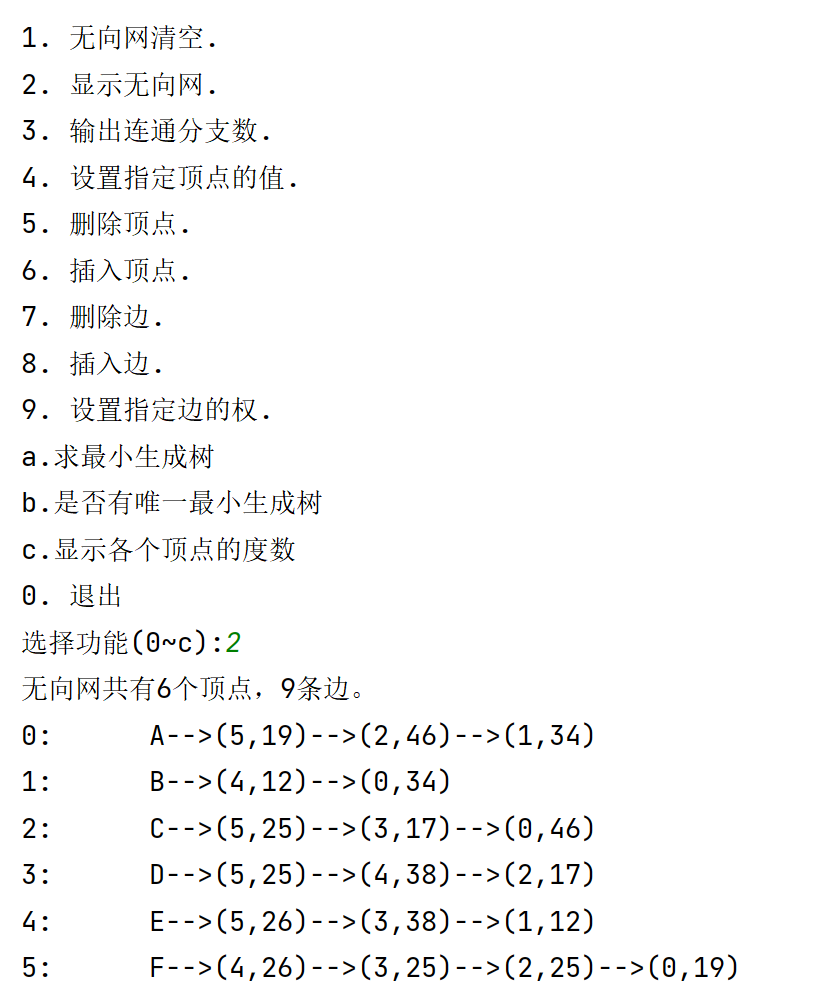




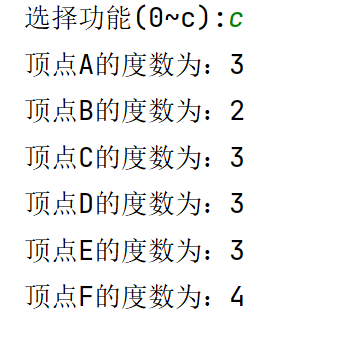


1. 实验结果

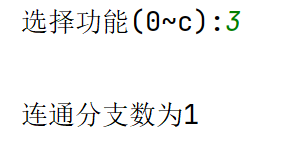
**显示用于测试的无向图：**



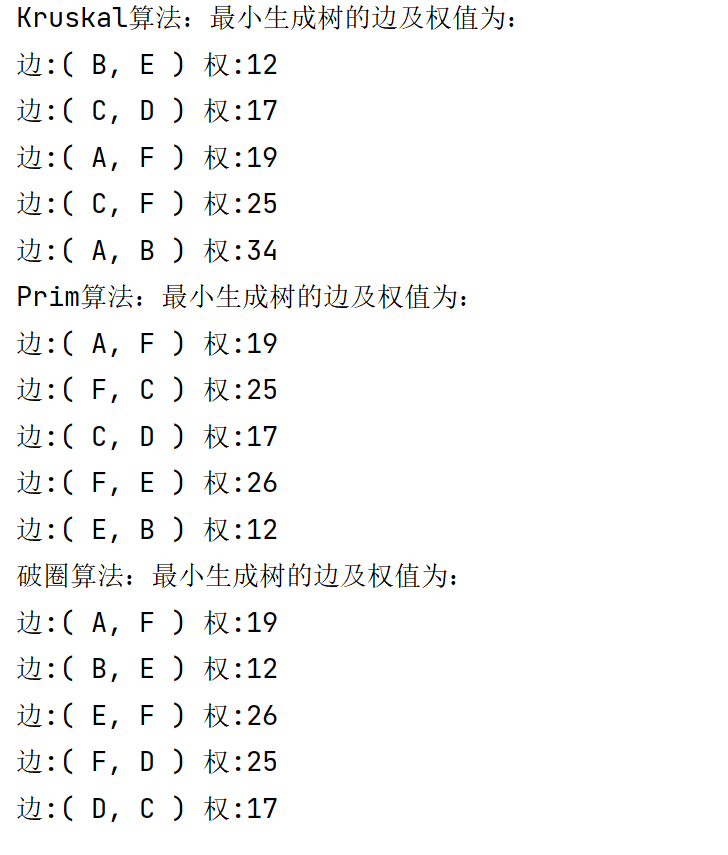
**统计各个顶点的度数：**



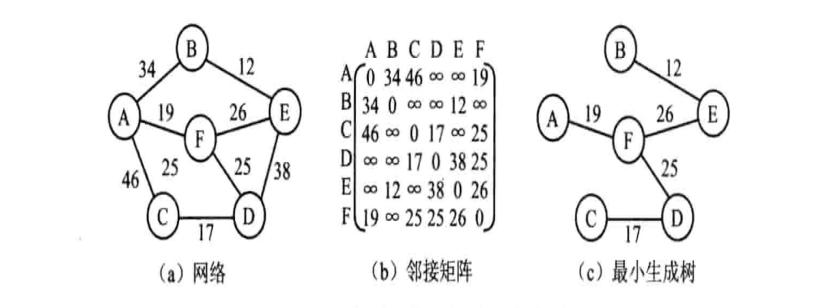
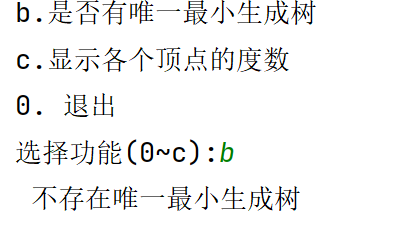
**求该图的连通分量数目**



**Kruskal、Prim、“破圈”算法求最小生成树**



**判断该图是否有唯一最小生成树**



实验现象与预想的实验现象一致。

1. 算法分析
2. CountDegree函数的时间复杂度为O(n),空间复杂度为O(1);
3. CountInDegree函数的时间复杂度为O(n+e),空间复杂度为O(n);
4. MiniSpanTreeBreakCycle函数时间复杂度为O(n2),空间复杂度为O(n);
5. hasUniqueMinTree函数的时间复杂度为O(n2), 空间复杂度为O(n);
6. n为顶点个数，e为边的数量
7. 总结与心得

在本次实验的过程中我们小组通过讨论，实践，查阅资料等方法完成了实验任务。通过实验，我们对于邻接表的基本概念以及其基本功能的实现有了更深入的理解。此外，我们还复习了最小堆，并查集等并运用到了解决问题中，学习了求最小生成树的其他算法，如“破圈法”，并亲自编写代码将其实现。总之，通过此次实验，我们提高了算法能力，增加了数据结构知识。

**四、分工说明**

李昀哲：算法设计，代码编写，ppt制作

唐铭锋：算法设计，代码编写，撰写报告

刘沛根：算法设计，代码编写，撰写报告

李正宇：算法设计，代码编写，ppt制作