

哈尔滨工业大学计算机科学与技术学院

2017 年秋季学期《软件工程》

Lab 1: 结对编程

姓名	班级/学号	联系方式
彭诗笑	4 班/1150310415	18845787871
龚玉贤	4 班/1150310422	18846142095

目 录

1	实验要求.....	1
2	待求解问题描述与数学模型.....	1
3	算法与数据结构设计.....	2
3.1	设计思路与算法流程图.....	2
3.2	数据结构设计.....	9
3.3	算法时间复杂度分析.....	9
4	实验与测试.....	9
4.1	读取文本文件并展示有向图.....	10
4.2	查询桥接词.....	13
4.3	根据桥接词生成新文本.....	14
4.4	计算最短路径.....	14
4.5	随机游走.....	15
5	编程语言与开发环境.....	17
6	结对编程.....	17
6.1	分组依据.....	17
6.2	角色切换与任务分工.....	17
6.3	工作照片.....	18
6.4	工作日志.....	19
7	小结.....	19

1 实验要求

练习结对编程，体验敏捷开发中的两人合作；

两人一组，自由组合；

使用一台计算机共同编码，完成实验要求；

在工作期间两人的角色至少切换 6 次；

使用 java+eclipse 编程，练习对 java 基本算法和数据结构的应用：

开发一个 java 命令程序，实现从文本中读取数据并根据要求生成图结构，输出该图结构，并在其上进行一系列计算操作，实时展示各操作的结果。

2 待求解问题描述与数学模型

输入 1：用户提供文本并输入或选择文本文件的位置。

输入 2：用户输入任意两个图中的英文单词。

输入 3：用户输入一段新的文本。

输入 4：用户输入任意两个图中的英文单词。

输出 1：输出用户所提供文本生成的有向图；

输出 2：输出用户所输入单词的桥接词或反馈无桥接词。

输出 3：将桥接词插入用户输入的新文本的两个单词之间并输出。

输出 4: 输出用户提供单词的最短路径或显示不可达。

输出 5: 随机游走并将遍历的节点生成文本存入磁盘。

3 算法与数据结构设计

3.1 设计思路与算法流程图

设计思路:

对每一个函数独立的思考设计, 然后合在一起形成一个完整的程序, 最后设计 GUI 框架, 并将程序放入框架中, 实现一个可视化的程序。

文件读取及处理: 利用 JAVA 的文件读取先将内容存到一个字符串中, 然后对字符串进行正则匹配消除干扰的字符, 最终得到一个标准的字符串。

生成展示图: 在这里考虑到后面的最短路径, 因此采用了邻接矩阵的方式来存储图, 因此在这里生成两个哈希表, HashMap1 键值分别存的是字符串(由第一个函数得到的字符串按照空格切割成一个字符串数组)和在哈希表的位置, HashMap2 则与 HashMap1 相反, 这是为了便于后续的操作。由此邻接矩阵建立完毕, 然后调用第三方库 graphViz 进行作图生成图片。

桥接词: 因为采用邻接矩阵的存储方式, 因此首先获得用户的输入, 将其在哈希表中的位置通过哈希表获得(分别为 ch, ch2), 对 G[ch][i] 进行遍历, 找到一个不为权值不为 0 的点 p, 对 G[p][ch2] 判断是否权值不为 0. 若不为 0 则将 p 对应的字符串加入到桥接词字符串 blank 中, 若为 0 则跳过。

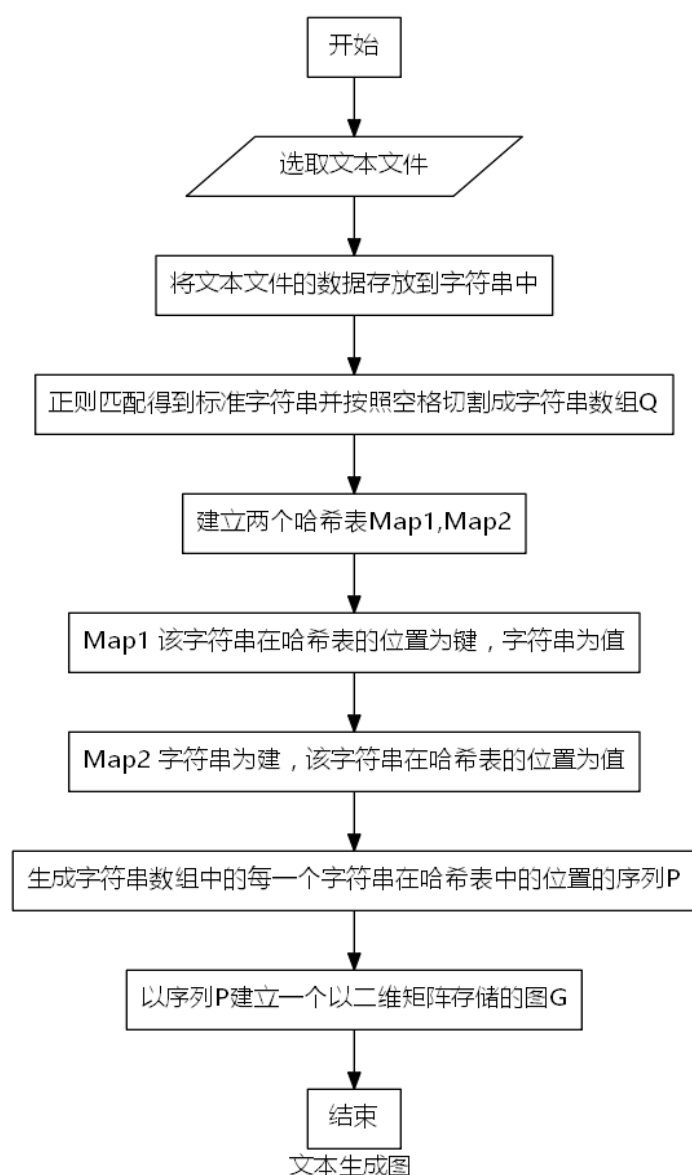
桥接词文本: 调用一个找桥接词的函数, 对输入的文本按照空格切割得到一个字符串数组, 然后循环, 每次都调用找桥接词函数, 如果有桥接词则加入两者中间再加入 blank 字符串, 没有则直接加入新的字符串 blank, 最后返回 blank 字符串。

最短路径: 运用 floyd 算法得到任意两点间的最短路径, 然后调用 graphViz 进行作图, 得到一个带有特殊标记的路径。

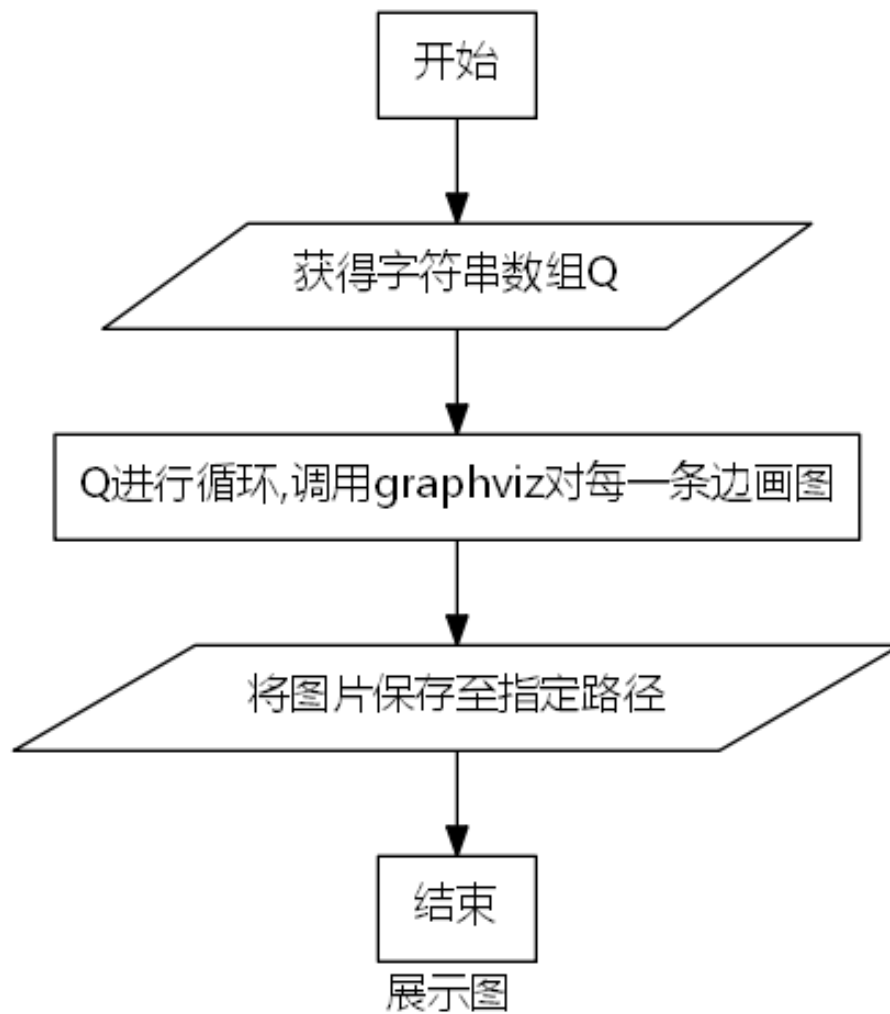
随机游走：随机产生一个数(在有效范围内)，生成一个路径矩阵，如果路径 $H \rightarrow I$ 走过则 $P[H][I]$ 则变为 0，每次游走都会判断是否该路径走过，同时如果没有出口则终止循环。每次找到一条边则输出，然后用户判定是否继续，如果停止则将路径字符串 blank 写入文本文件 tete.txt;

算法流程图：

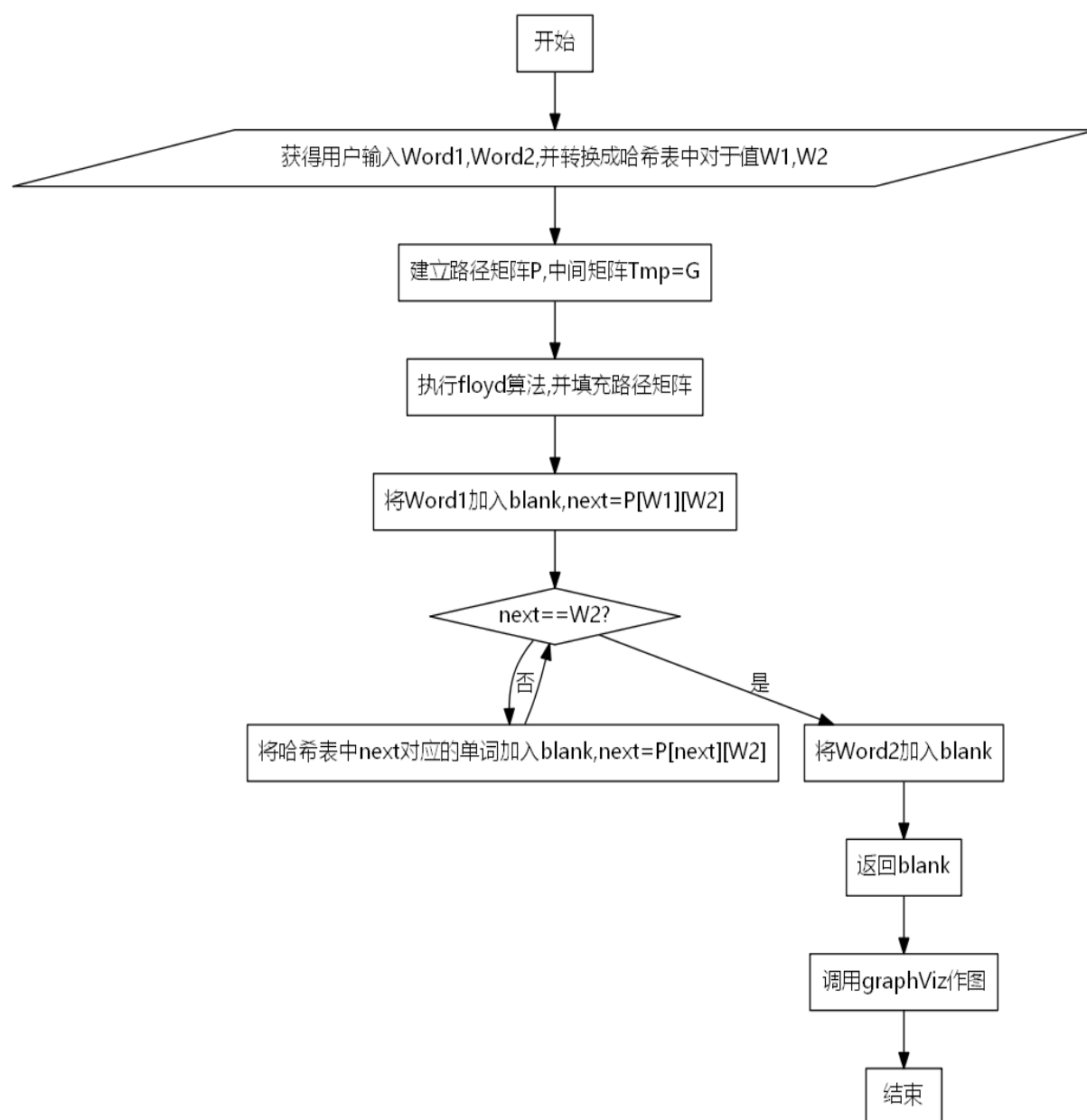
文本生成图：



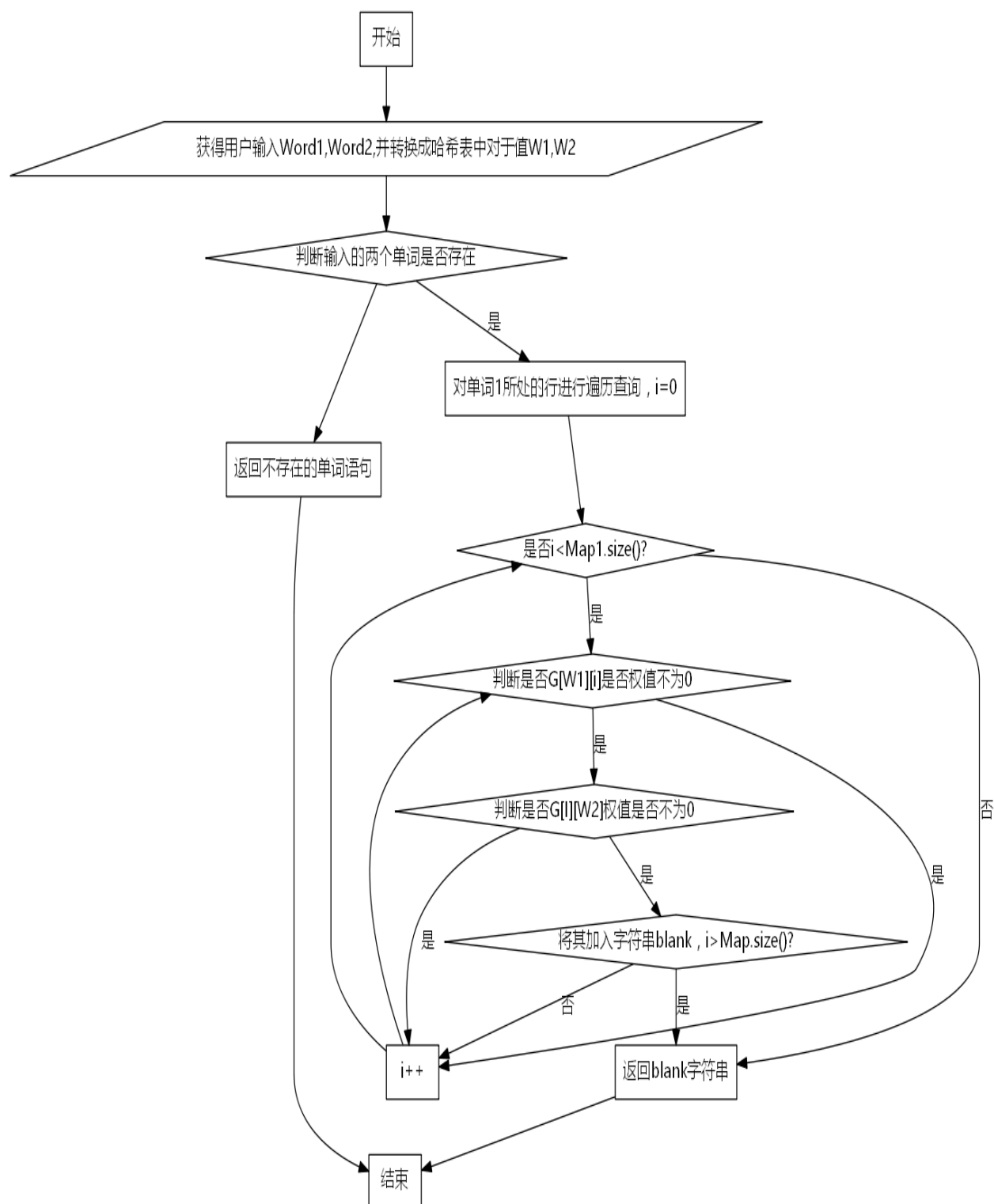
展示生成的图：



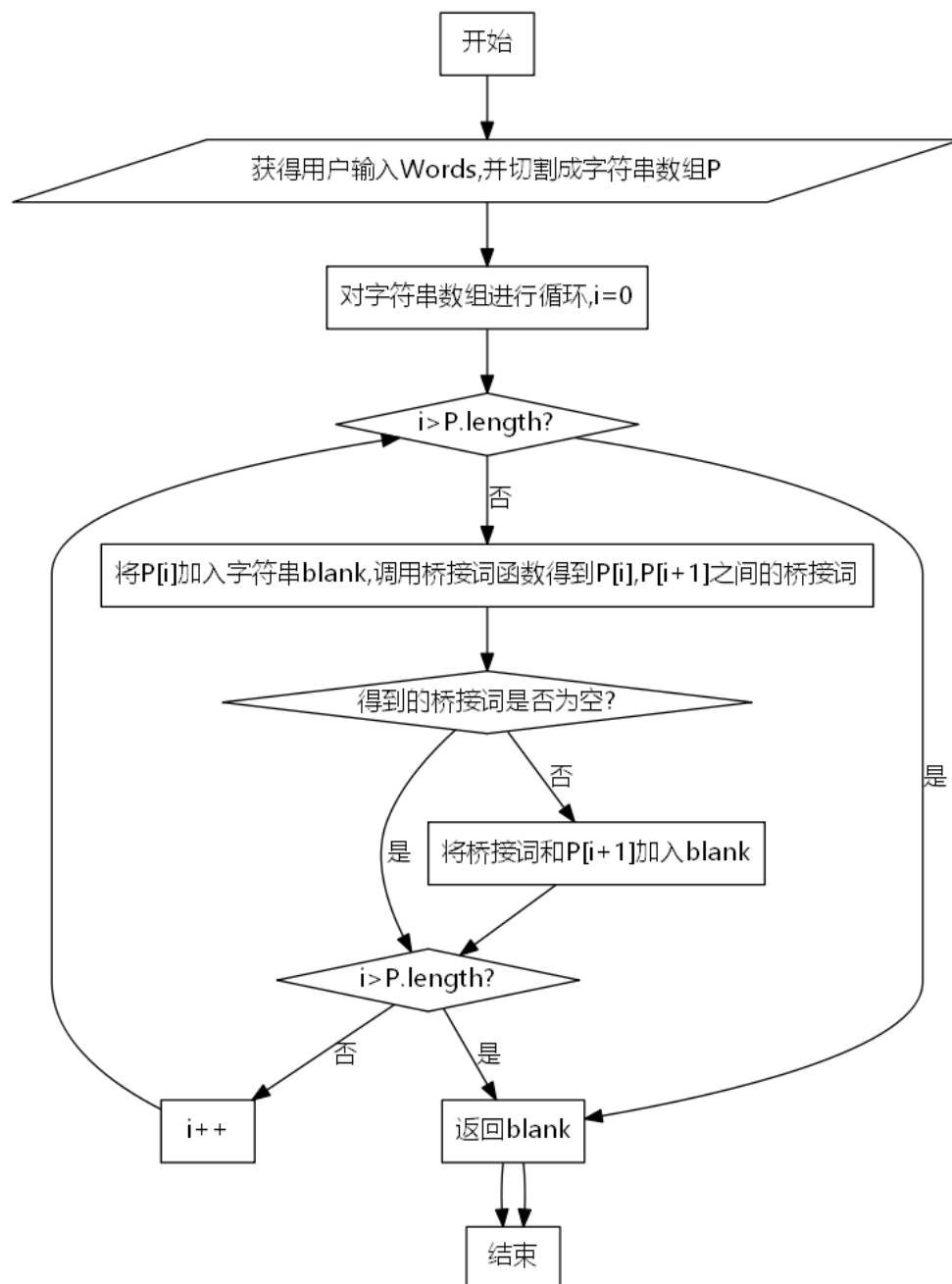
最短路径:



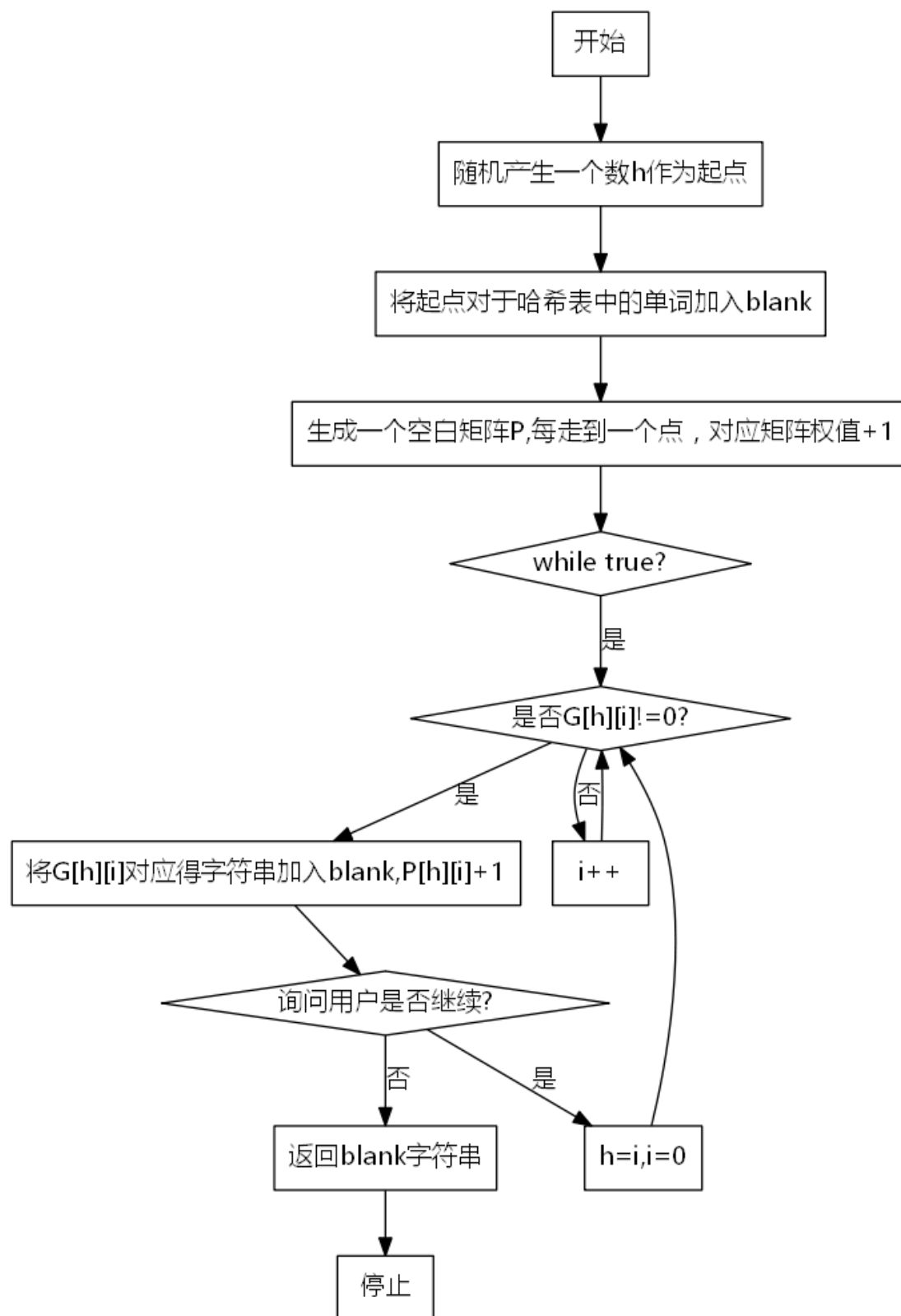
找桥接词:



在新文本中加入桥接词：



随机游走：



3.2 数据结构设计

图:图是由结点的有穷集合 V 和边的集合 E 组成。其中, 为了与树形结构加以区别, 在图的结构中常常将结点称为顶点, 边是顶点的有序偶对, 若两个顶点之间存在一条边, 就表示这两个顶点具有相邻关系。本次实验采用了二维数组的方式存储图, 即邻接表。

散列表:若结构中存在关键字和 K 相等的记录, 则必定在 $f(k)$ 的存储位置上。由此, 不需比较便可直接取得所查记录。称这个对应关系 f 为散列函数, 按这个思想建立的表为散列表。

3.3 算法时间复杂度分析

文本生成图:在这里用到一个循环读取文件内容, 同是建立两个哈希表, 最后生成以邻接矩阵存储的图, 因为已经建立哈希表, 因此可以通过哈希表的到一个数字序列(即字符串在哈希表中的位置), 因此一个循环即可, 时间复杂度 $O(n)$ 。

展示图:调用第三方的库 `graphViz` 进行画图, 一次循环即可, 时间复杂度 $O(n)$ 。

查询桥接词:起点终点确定, 因此只需要对起点那行进行遍历找不为 0 的点, 再对不为 0 的点那列进行判定即可, 时间复杂度 $O(n)$ 。

根据桥接词生成新文本:外层对文本的每一个单词进行循环, 内层需要调用桥接词函数, 因此时间复杂度 $O(n^2)$ 。

最短路径:这里需要用到 floyd 算法, 时间复杂度为 $O(n^3)$ 。

随机游走:因为每次游走需要用户的判定, 加上只是对某行进行查询, 时间复杂度 $O(n)$ 。

4 实验与测试

设计 1 个至少包含 50 个单词的输入文本文件, 使之可覆盖本题目中关于输入文件和功能的各种特殊情况, 作为你开发的程序的输入。

针对在有向图上操作的每项功能, 为其设计各种可能的输入数据。输入数据的数量不限, 以测试程序的充分性为评判标准(下面各节中的表格的行数请自行扩展)。

记录程序的输出结果, 判断输出结果是否与期望一致, 并记录程序运行截图。

4.1 读取文本文件并展示有向图

文本文件中包含的内容:

The way @ she came into place I knew miss like.

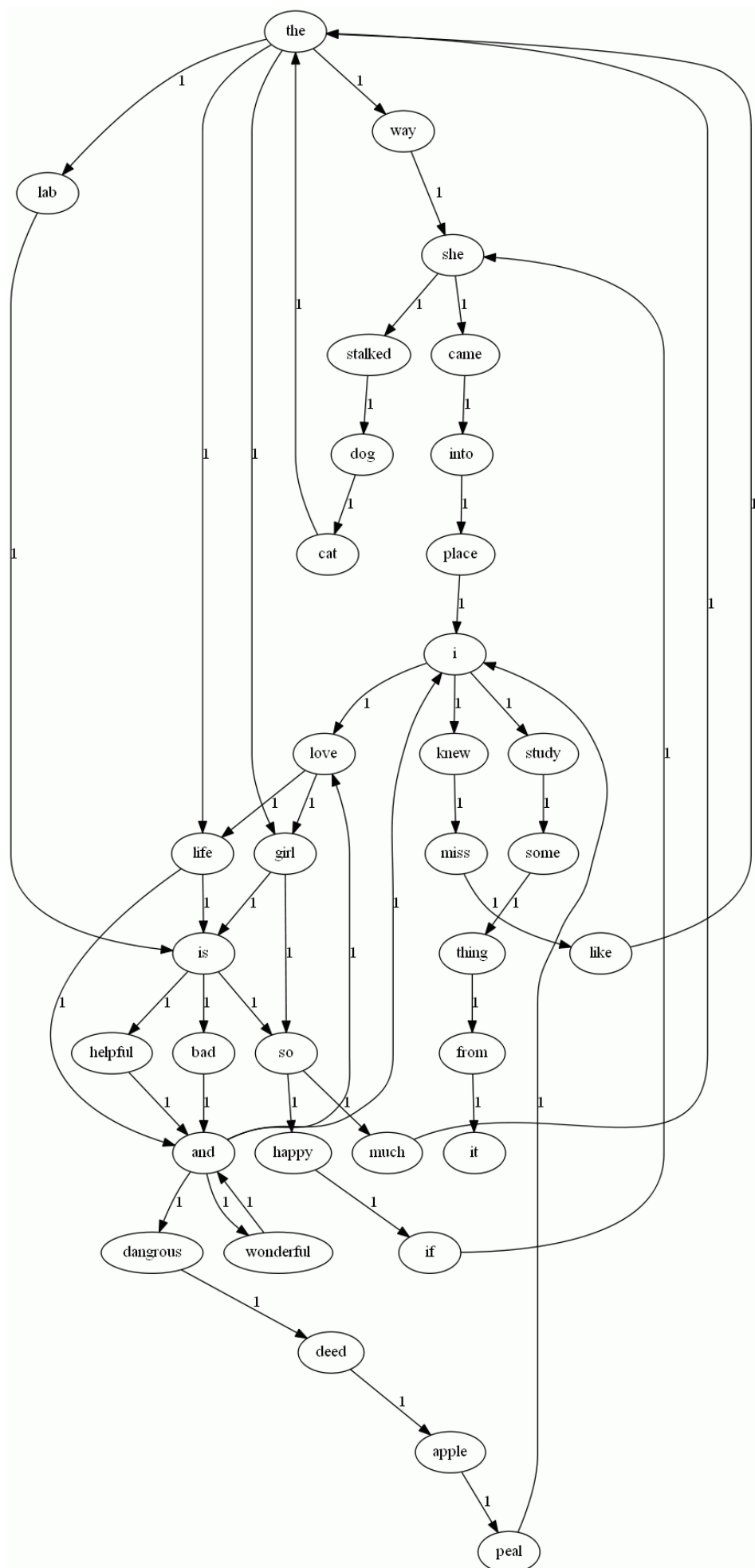
The life #is so happy if she stalked dog cat.

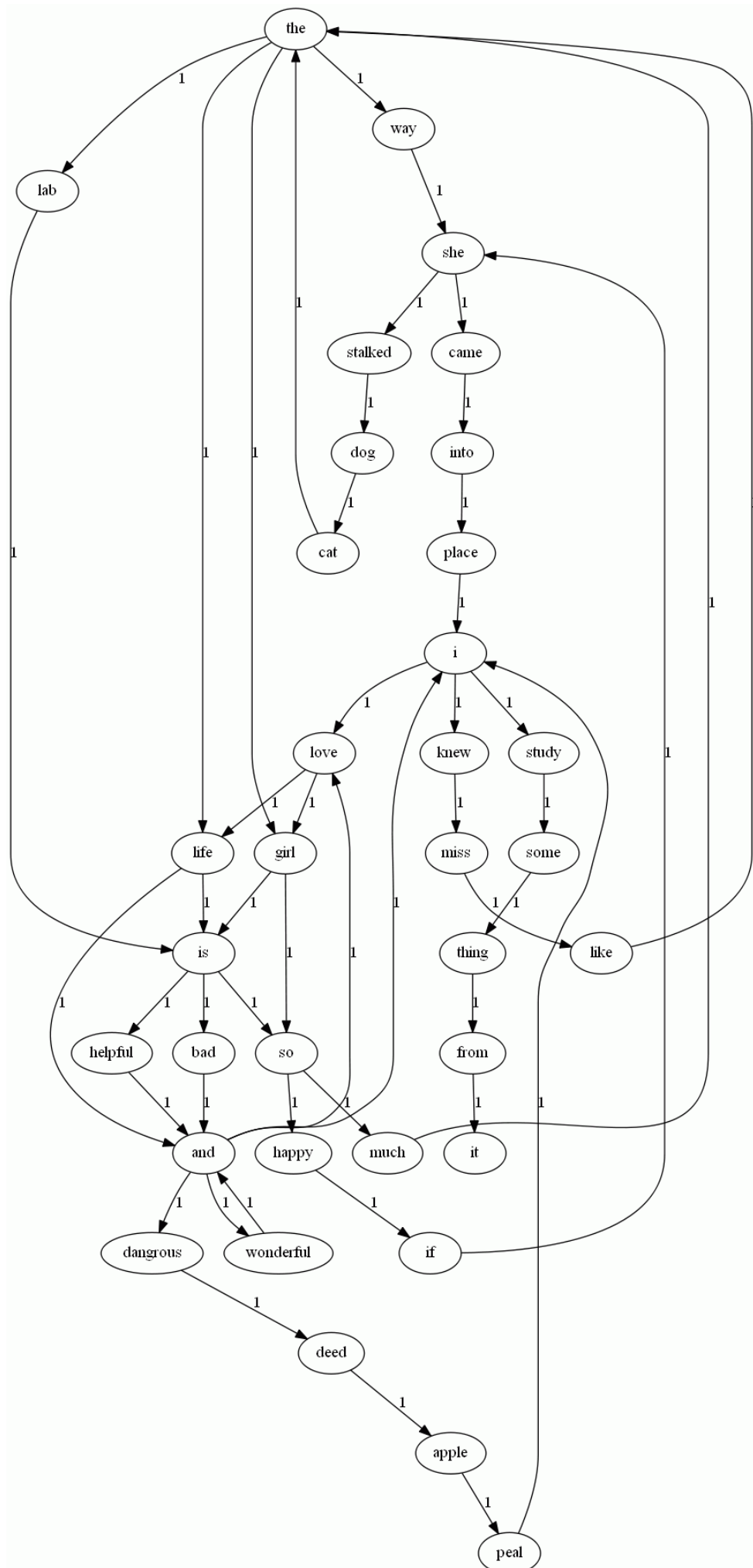
The girl is bad and dangrous deed apple peal.

I love life and love girl so much!

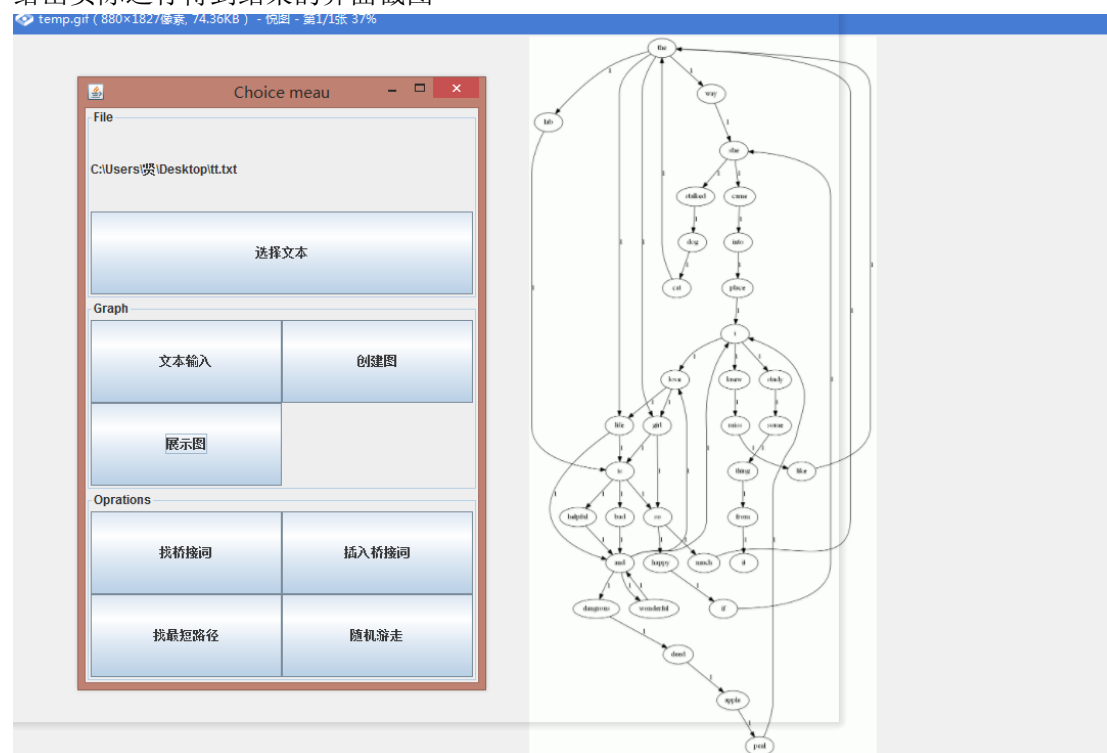
The lab is helpful and wonderful and I study some thing from it.

下面第一张为手工计算所得的图，第二张为代码所画的图，二者一致，证明所写程序满足题目要求。





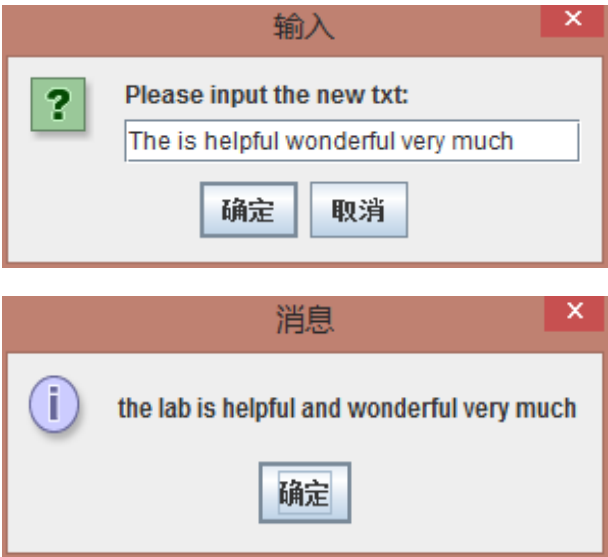
给出实际运行得到结果的界面截图





4.3 根据桥接词生成新文本

序号	输入（一行文本）	期望输出	实际输出	运行是否正确
1	The is helpful wonderful very much	The lab is helpful and wonderful very much	The lab is helpful and wonderful very much	是



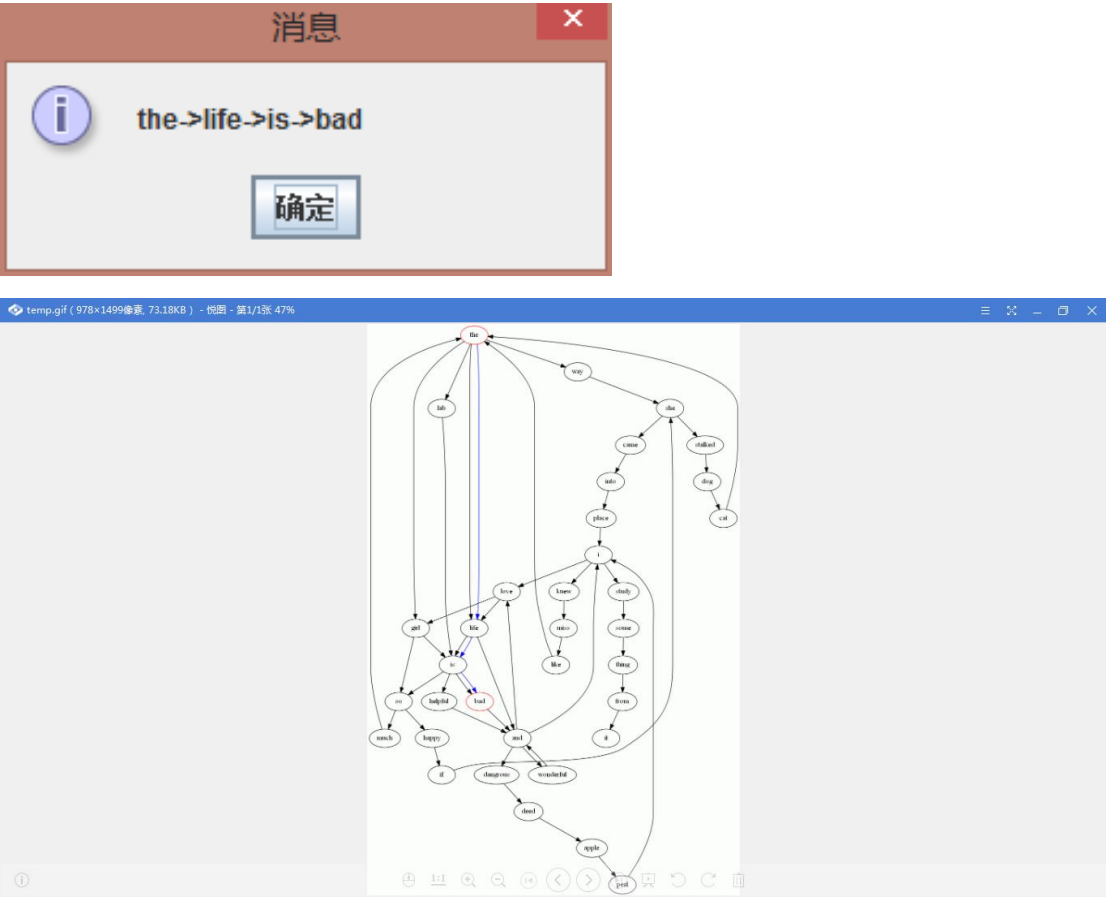
..

4.4 计算最短路径

序号	输入（两个单词、或一个单词）	期望输出	实际输出	运行是否正确
----	----------------	------	------	--------

1	the bad	the->life->is->bad	the->life->is->bad	是
---	---------	--------------------	--------------------	---

给出实际运行得到结果的界面截图。

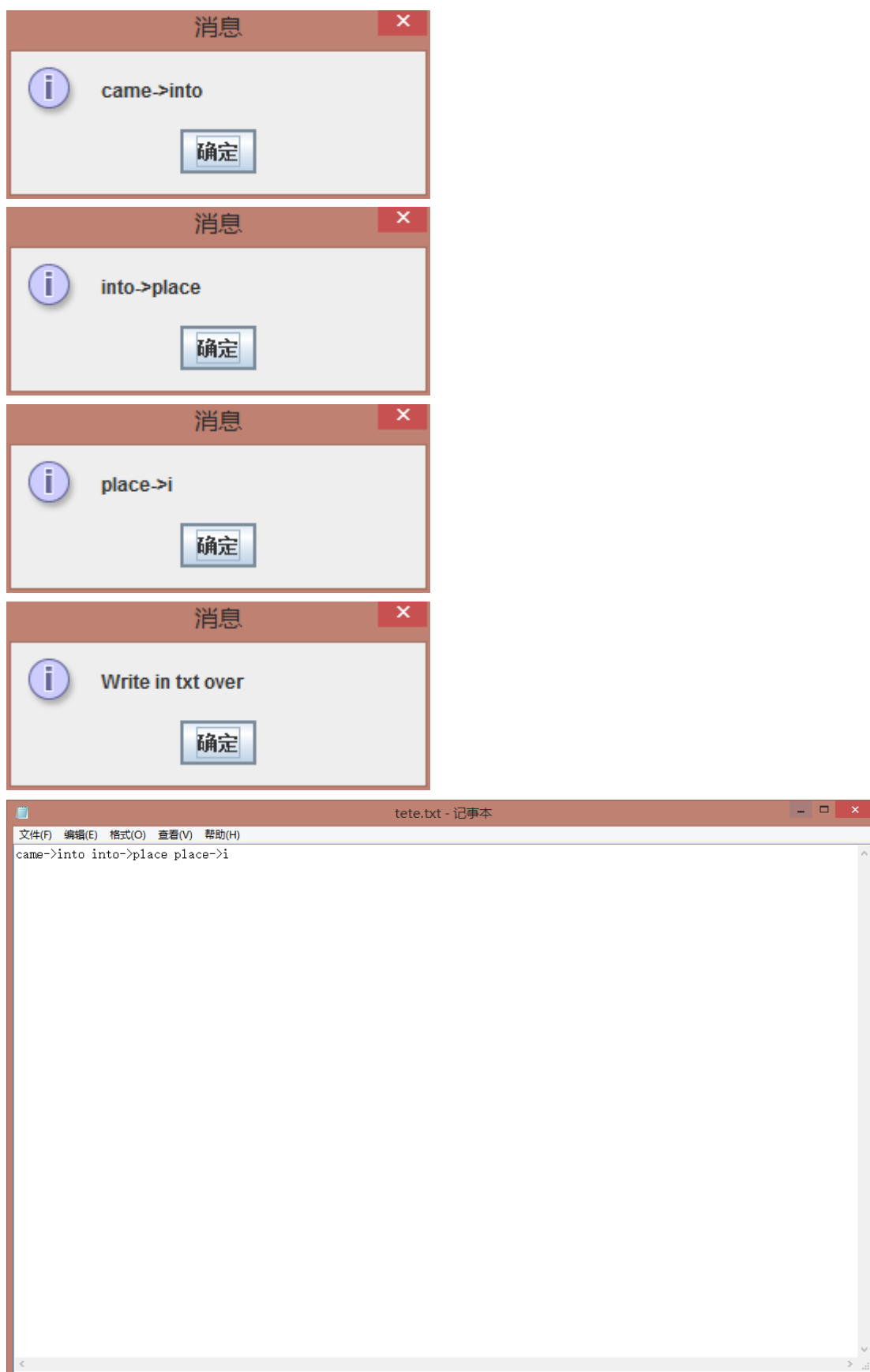


4.5 随机游走

该功能无输入，让你的程序执行多次，分别记录结果。

序号	实际输出	程序运行是否正确
1	came->into into->place place->i	是
2	and->i i->knew knew->miss miss->like like->the the->way way->she she->came came->into into->place place->i i->love love->life life->is is->so so->happy happy->if if->she she->stalked stalked->dog dog->cat cat->the the->life life->and and->dangrou s dangrou->deed deed->apple apple->peal peal->i i->study study->some some->thing thing->from from->it	是
3	way->she she->came came->into into->place place->i i->knew knew->miss miss->like like->the the->way	是

给出实际运行得到结果的界面截图。
仅序号一有截图（图太多就不全放了）



5 编程语言与开发环境

Java DK: java version "1.8.0_144"

Eclipse IDE: Oxygen Release (4.7.0)

Graphviz -2.38

6 结对编程

6.1 分组依据

彭诗笑的特点：做事认真仔细；算法数据结构扎实；执行力强。

我的特点：勤奋认真；善于沟通交流；不够细致。

我们可以优势互补，共同学习进步

6.2 角色切换与任务分工

该表格可自行增加更多的行。

日期	时间(HH:MM -- HH:MM)	“驾驶员”	“领航员”	本段时间的任务
9-15	14: 00-17: 00	彭诗笑	龚玉贤	建立图
9-16	18: 00-21: 00	龚玉贤	彭诗笑	找桥接词
9-17	15: 00-18: 00	彭诗笑	龚玉贤	在文本中加桥接词
9-17	19: 00-21: 00	龚玉贤	彭诗笑	随机游走
9-18	14: 30-15: 00	彭诗笑	龚玉贤	展示图
9-18	15: 00-18: 00	龚玉贤	彭诗笑	最短路径
9-19	14: 00-17: 00	彭诗笑	龚玉贤	GUI用户界面

6.3 工作照片



6.4 工作日志

由领航员负责记录，记录结对编程期间的遇到的问题、两人如何通过交流合作解决每个问题的。可增加表格的行。

日期/时间	问题描述	最终解决方法	两人如何通过交流找到解决方法
9-18	随机游走时部分路径未能游走到本应能走到的结点	找到函数中出现的问 题，将其修改正确	先多次测试文本，找到错误的点，再不断调整函数，进行调试
9-19	图形的展示方法不够理想	利用 Graphviz 软件制图	互相交流方法，并查阅资料，选取简单有效的方法

7 小结

结对编程这个概念还是第一次接触，通过这次实验发现它确实可以帮助我们共同学习，提高我们的编程能力，交流能力以及团队合作能力。

我能明显感觉到两人编程效率和质量的提升，我们可以相互交流，学习彼此在编程中所具有的优秀品质，也能发现自己的不足，并努力去提高，同时还增强了两人的团队协作能力。

结对编程能让我们注意到更多的细节，交换对代码的理解，也能让我们相互竞争，展现自己的编程能力，但可能会存在理解和使用方法难以统一，工作节奏不协调的问题。

我认为倡导结对编程是因为相较于一个人编程，它具有很多优势。举个例子：即便是最复杂的设计，只要是你自己想出来的，你都觉得它简单无比，里面充满了直白且显而易见的理由。可惜不幸的是，我们要的简单，是对 team 里所有人的简单。如果你的 pair 不能理解你的设计，那么说明你的设计复杂了；如果你们两个人懂，但是 switch pair 的时候，换过来的人不懂，说明你的设计复杂了。在这个相互交流的过程中能够互相提高，并且提升效率。

我期待着下一次的结对编程，既可以在小伙伴面前露一手，又能在此过程中学到不少知识，并且提高自己的编程能力。