

**2018年春季学期  
计算机学院大二软件构造课程**

**Lab 4实验报告**

|  |  |
| --- | --- |
| 姓名 | 郭潇 |
| 学号 | 1160300521 |
| 班号 | 03005 |
| 电子邮件 | Wqsnzszw@163.com |
| 手机号码 | 18523735573 |

**目录**

[1 实验目标概述 1](#_Toc514612714)

[2 实验环境配置 1](#_Toc514612715)

[3 实验过程 1](#_Toc514612716)

[3.1 Error and Exception Handling 1](#_Toc514612717)

[3.1.1 针对输入文本文件的异常/错误处理 1](#_Toc514612718)

[3.1.2 针对输入图操作指令的异常/错误处理（可选） 1](#_Toc514612719)

[3.2 Assertion and Defensive Programming 1](#_Toc514612720)

[3.2.1 checkRep()检查invariants 1](#_Toc514612721)

[3.2.2 Assertion保障pre-/post-condition 1](#_Toc514612722)

[3.3 Logging 1](#_Toc514612723)

[3.3.1 写日志 1](#_Toc514612724)

[3.3.2 日志查询 1](#_Toc514612725)

[3.4 Testing for Robustness and Correctness 1](#_Toc514612726)

[3.4.1 Testing strategy 1](#_Toc514612727)

[3.4.2 测试用例设计 1](#_Toc514612728)

[3.4.3 测试运行结果与覆盖度报告 1](#_Toc514612729)

[3.5 FindBugs tool（可选） 1](#_Toc514612730)

[3.6 Debugging 1](#_Toc514612731)

[3.6.1 待调试程序 1](#_Toc514612732)

[3.6.2 理解待调试程序的过程 1](#_Toc514612733)

[3.6.3 发现并定位错误的过程 1](#_Toc514612734)

[3.6.4 如何修正错误 1](#_Toc514612735)

[3.6.5 结果 1](#_Toc514612736)

[4 实验进度记录 1](#_Toc514612737)

[5 实验过程中遇到的困难与解决途径 1](#_Toc514612738)

[6 实验过程中收获的经验、教训、感想 1](#_Toc514612739)

# 实验目标概述

根据实验手册简要撰写。

# 实验环境配置

简要陈述你配置本次实验所需环境的过程，必要时可以给出屏幕截图。

特别是要记录配置过程中遇到的问题和困难，以及如何解决的。

在这里给出你的GitHub Lab4仓库的URL地址（Lab4-学号）。

# 实验过程

请仔细对照实验手册，针对每一项任务，在下面各节中记录你的实验过程、阐述你的设计思路和问题求解思路，可辅之以示意图或关键源代码加以说明（但千万不要把你的源代码全部粘贴过来！）。

## Error and Exception Handling

### 针对输入文本文件的异常/错误处理

* 1.文件中存在不符合语法规则的语句
* 错误类型：ExceptionofInput
* 错误原因：指令缺少分量；关于边是否有向的信息错误；
* 处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；
* 2. 边或点使用的类型在未提前定义
* 错误类型：ExceptionofInput
* 处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；

3. 为节点定义的属性的数目与特定应用的图的约束不符

* 错误类型：ExceptionofInput
* 错误原因：输入信息为包含节点的所有属性
* 处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；
* 4. 在某种类型的图应用中引入了不应出现的节点或边的类型
* 错误类型：ExceptionofInput
* 处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；

5. 在无向图中引入了有向边

* 错误类型：ExceptionofUndirection
* 处理方式：将该边变为无向边，继续向下执行

6. 在有向图中引入了无向边

* 错误类型：ExceptionofDirection
* 处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；

7. 在单重图中存在了多重边

* 错误类型：ExceptionofUnproperEdge
* 处理方式：忽略该条件且不加入该边，保留之前的边，继续向下执行。

8. 在不应存在边的两个节点之间存在了边

* 错误类型：ExceptionofUnproperEdge
* 处理方式：忽略该条件，不加入非法边，继续向下执行。

9. 在不应存在超边的图中加入了超边

* 错误类型：ExceptionofUnproperEdge

处理方式：忽略该条件，不加入非法边，继续向下执行。

10. 某超边中包含的节点数小于 2

* 错误类型：ExceptionofUnproperEdge

处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；

11. 带权边的权值不符合应用要求

* 错误类型：ExceptionofUnproperEdge

处理方式：提示错误信息，按ERROR级别写入log；提示选择下一个文件；

12. 多个节点的 label 重复、多个边的 label 重复

* 错误类型：ExceptionofLabel

处理方式：按Warn级别写入log；应通过自动修改某些 label来除重复；输出修改后的label名；

### 针对输入图操作指令的异常/错误处理（可选）

覆盖实验手册3.1节中(2)列出的各项任务。

## Assertion and Defensive Programming

### checkRep()检查invariants

vertex类型：

1. Word类：checkRep确定label不为空。
2. Person、Actor、Director类：checkRep确定label不为空，年龄大于0且性别为F或M。
3. Computer、Router、Server类：checkRep确定label不为空，IP地址用“.”分割为四部分，每部分的 取值范围为[0,255]。
4. Movie类：checkRep确定label和拍摄国家不为空，上映年份属于[1900, 2018]，评分属于 0-10且为二位小数。

Edge类型：

1. WordNeighborhood类：checkRep确定label不为空；source和tagert不为空；source和tagert类型都为Word；weight大于0；
2. FriendTie、ForwardTie、CommentTie类：checkRep确定label不为空；source和tagert不为空；source和tagert类型都为Person；weight大于0且小于等于1；
3. NetworkConnection类：checkRep确定label不为空；source和tagert不为空；source和tagert类型为Computer、Router、Server但不能都为Computer或Server；weight大于等于0；
4. MovieActorRelation类：checkRep确定label不为空；source和tagert不为空；source和tagert类型为Movie和Actor；weight大于0；
5. MovieDirectorRelation类：checkRep确定label不为空；source和tagert不为空；source和tagert类型为Movie和Director；
6. SameMovieHyperEdge类：checkRep确定label不为空；vertices的个数大于1；

Graph类型：

1. RI为满足图中顶点不为null，边不形成自环
2. AF为不形成自环边和不为null的顶点

### Assertion保障pre-/post-condition

vertex类型：

确定vertex的label不为null，附加信息例如性别只能为“F”或“M”，ip地址必须符合给定格式；weight也为该类型顶点的要求范围。

Edge类型：

确定edge的label不为null，加入的顶点不能为null，权值也为该类型的范围。

Graph类型：

确定加入的顶点和边不为null；

## Logging

### 写日志

配置：将异常和操作分为三个等级：基本的操作为info等级；提示异常但忽略的为warn等级，提示异常进行处理的为error等级。Info以上等级的写入log.log文件；warn以上等级在控制台上输出错误并写入warn.log文件；ERROR  
以上等级写入error.log文件

格式：

1. INFO等级对应基本操作，格式为：
2. WARN等级对应提示信息但忽略的异常，格式为：
3. ERROR等级对应提示信息并进行处理的异常，格式为：

### 日志查询

查询方式：

1. 按照等级查询：INFO、WARN和ERROR等级的错误保存在不同的文件中，选择需要查询的最低等级，则按照简要格式输出相应文件内容。
2. 按照时间段查询：输入合法时间段，从INFO文件中读取每条信息与给定信息比较，若属于给定时间段，则输出简要格式信息。

## Testing for Robustness and Correctness

### Testing strategy

Vertex类：

1. Word类：chekRep：label是否为空；equals：label是否相同；类是否相同。

1. Person、Actor、Director类：chekRep：label是否为空；性别是否符合要求；年龄是否符合要求；equals：label是否相同；类是否相同。
2. Computer、Router、Server类：chekRep：label是否为空；ip地址格式是否符合要求；equals：label是否相同；类是否相同。
3. Movie类：chekRep：label是否为空；拍摄国家是否为空；拍摄年份是否符合要求；equals：label是否相同；类是否相同。

Edge类

1. WordNeighborhood类：checkRep：label是否为空；source和tagert是否为空；source和tagert类型是否为Word；weight大于0；equals：source和target是否都相同；类是否相同
2. FriendTie、ForwardTie、CommentTie类：checkRep：label是否为空；source和tagert是否为空；source和tagert类型是否为Person；weight是否大于且小于等于1；source和target是否相同；equals：source和target是否都相同；类是否相同
3. NetworkConnection类：checkRep：label是否为空；source和tagert是否为空；source和tagert类型是否为相应类型；weight是否大于0；source和target是否相同；equals：source和target是否都相同；类是否相同
4. MovieActorRelation类：checkRep：label是否为空；source和tagert是否为空；source和tagert类型是否为Movie和Actor；weight是否大于0；source和target是否相同；equals：source和target是否都相同；类是否相同
5. MovieDirectorRelation类：checkRep：label是否为空；source和tagert是否为空；source和tagert类型是否为Movie和Director； source和target是否相同；equals：source和target是否都相同；类是否相同
6. SameMovieHyperEdge类：checkRep：label是否为空；vertices类型是否为Actor；vertices大小是否大于1;

Graph类：

1. addVertex操作按照graph的大小为0、1、n；vertex是否在图中划分
2. addEdge操作按照graph的大小为0、1、n；weight的大小是否为0；target，source是否在graph中；加入的edge是否是new edge划分
3. removeVertex操作按照graph的大小为0、1、n；vertex是否在图中;
4. removeEdge操作按照edge的数量为0、1、n；edge是否在图中;
5. vertices操作按照graph的大小为0、1、n划分
6. Edge操作按照edge的数量为0、1、n划分
7. sources操作按照graph的大小为0、1、n；edge的数量为0、1、n划分
8. targets操作按照graph的大小为0、1、n；edge的数量为0、1、n划分

Exception类：

* 1.文件中存在不符合语法规则的语句:分类为：指令类型：边指令，点指令，超边指令；图类型：四种图；边是否有向信息：正确，错误；
* 2. 边或点使用的类型在未提前定义: 分类为：指令类型：边指令，点指令，超边指令；图类型：四种图；是否定义：定义，未定义；
* 3. 为节点定义的属性的数目与特定应用的图的约束不符：分类为：图类型：四种图；属性的数目是否完全；完全：不完全；
* 4. 在某种类型的图应用中引入了不应出现的节点或边的类型：分类为：指令类型：边指令，点指令，超边指令；图类型：四种图；可否出现；可以，不可以：

5. 在无向图中引入了有向边：图类型：两种图；边的方向是否正确；是，否：

6. 在有向图中引入了无向边：图类型：两种图；边的方向是否正确；是，否：

7. 在单重图中存在了多重边：图类型：四种图；多重边是否存在；是，否：

8. 在不应存在边的两个节点之间存在了边： 图类型：两种图；顶点类型间是否可存在边；是，否：

9. 在不应存在超边的图中加入了超边：图类型：三种图；该图是否可存在超边；是，否：

10. 某超边中包含的节点数小于 2：超边的顶点数：大于等于2，小于2；

11. 带权边的权值不符合应用要求：图类型：三种图；边的权值是否符合要求；是，否：

* 12. 多个节点的 label 重复、多个边的 label 重复：指令类型：边指令，点指令；图类型：四种图；边的权值是否符合要求；label是否重复；

### 测试用例设计

Vertex类：

1. Word类：label是否为空；checkrep：label为空；equals：label和类都相同；label相同但类不同；label不同但类相同；

1. Person、Actor、Director类：checkrep：label为空；性别和年龄都符合要求；性别符合要求但年龄不符合要求；年龄符合要求但性别不符合要求；equals：label和类都相同；label相同但类不同；label不同但类相同；
2. Computer、Router、Server类：checkrep：label为空；ip地址符合要求；ip地址不符合要求；equals：label和类都相同；label相同但类不同；label不同但类相同；
3. Movie类：checkrep：label为空；拍摄国家为null；拍摄年份不符合要求；拍摄年份不符合要求；equals：label和类都相同；label相同但类不同；label不同但类相同；

Edge类：

1. WordNeighborhood类： label为空；source和tagert的size为2；source和tagert类型不都为Word；weight小于0；equals：source和target是都相同且类相同；类不相同
2. FriendTie、ForwardTie、CommentTie类：label为空；source和tagert为空；source和tagert类型不为Person；weight大于1；source和target相同；equals：source和target都相同但类不同；类相同但source和target不同
3. NetworkConnection类：checkRep： source和tagert类型不为相应类型；weight小于0；source和target相同；source和target为不该出现类型；equals：equals：source和target都相同但类不同；类相同但source和target不同
4. MovieActorRelation类：checkRep： source和tagert类型不为相应类型；weight小于0；source和target相同；source和target为不该出现类型；equals：equals：source和target都相同但类不同；类相同但source和target不同；
5. MovieDirectorRelation类：checkRep： source和tagert类型不为相应类型；source和target相同；source和target为不该出现类型；equals：equals：source和target都相同但类不同；类相同但source和target不同；

6.SameMovieHyperEdge类：checkRep： vertices类型都为Actor但大小小于1；vertices大小大于1但类型不都为Actor;

Graph类：

对实验3中原有测试进行修改。

Exception类：

* 1.文件中存在不符合语法规则的语句:测试为：各类型图的点指令与边指令与错误的组合，共6个；
* 2. 边或点使用的类型在未提前定义: 测试为：各类型图加上点或边指令与错误的组合，共4个；
* 3. 为节点定义的属性的数目与特定应用的图的约束不符：测试为：各类型图加上点或边指令与错误的组合，共4个；
* 4. 在某种类型的图应用中引入了不应出现的节点或边的类型：测试为：各类型图加上点或边指令与错误的组合，共4个；

5. 在无向图中引入了有向边：各类型图与错误的组合，共2个；

6. 在有向图中引入了无向边：各类型图与错误的组合，共2个；

7. 在单重图中存在了多重边：各类型图与错误的组合，共4个；

8. 在不应存在边的两个节点之间存在了边： 各类型图与错误的组合，共3个；

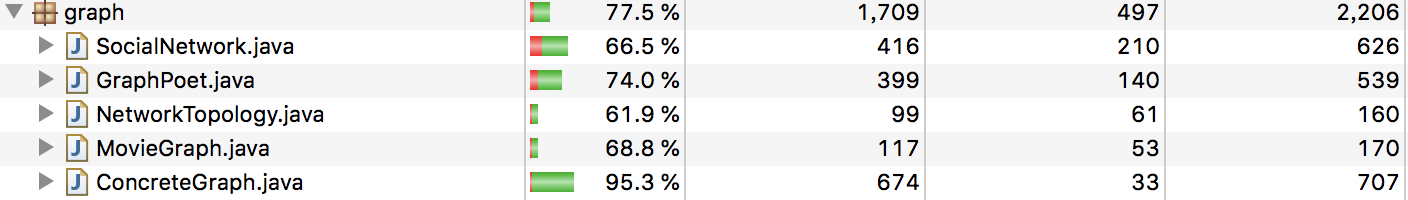
9. 在不应存在超边的图中加入了超边：各类型图与错误的组合，共3个

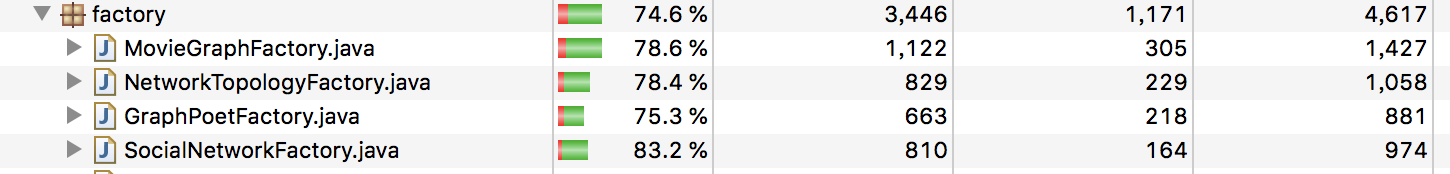
10. 某超边中包含的节点数小于 2：按是否错误分类，共2个；

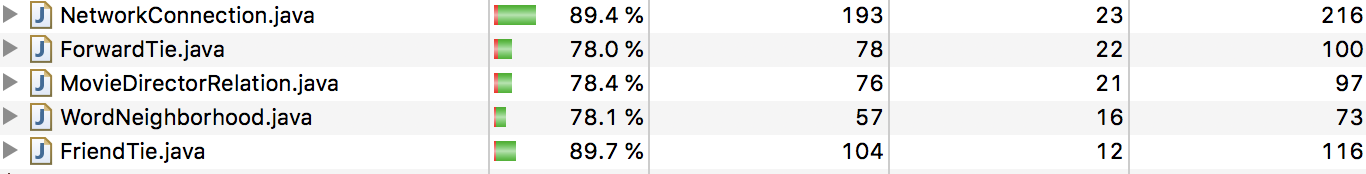
11. 带权边的权值不符合应用要求：各类型图与错误的组合，共3个

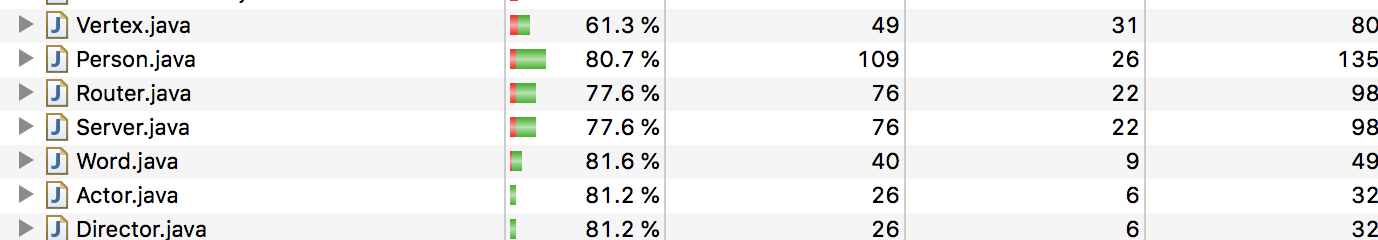
* 12. 多个节点的 label 重复、多个边的 label 重复：各类型图加上点或边类型指令与错误的组合，共8个

### 测试运行结果与覆盖度报告









## FindBugs tool（可选）

## Debugging

### 待调试程序

1. 计算器
2. 图形

### 理解待调试程序的过程

1. 计算器：运行程序，通过观察发现错误并定位错误产生位置并修改。
2. 图形：编译器提示子类的部分方法未实现，尝试运行程序也出现错误，根据提示逐步修改程序。

### 发现并定位错误的过程

1. 计算器：发现数字和运算符号不符合要求，于是检查输出数字和运算符号的for循环发现错误。
2. 图形：编译器提示了子类方法缺失，根据数组越界发现输出时的错误

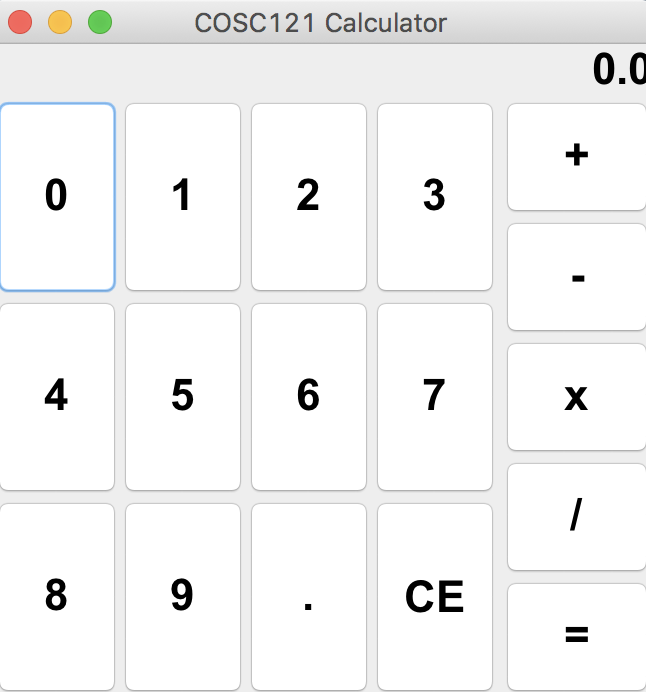
### 如何修正错误

1. 计算器：结合数字和运算符号的数量。确定for循环的正确结构并修改。
2. 图形：添加缺少子类的方法，修改输出时的界限。

### 结果

1. 计算机：

具有以下界面，且完善了计算器功能。



1. 图形：输出各图形信息正确。

# 实验进度记录

请尽可能详细的记录你的进度情况。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 计划任务 | 实际完成情况 |
| 05-07 | 13：45-15：30 | 阅读实验手册，理解实验要求 | 完成 |
| 05-09 | 18：30-21：30 | 编写异常类型ExceptionofInput相关内容 | 未完成 |
| 05-10 | 16：00-18：00 | 完成异常类型ExceptionofInput相关内容 | 完成 |
| 05-11 | 14：00-17：30 | 完成异常类型ExceptionofUnproperEdge内容 | 完成 |
| 05-11 | 19：00-21：30 | 根据实验手册，完善异常类型 | 完成 |
| 05-12 | 19：00-21：00 | 编写checkrep相关 | 发现实验3的不足，未完成 |
| 05-13 | 15：00-17：00 | 修改之前错误，继续编写checkrep相关 | 完成 |
| 05-14 | 13：45-15：30 | 完善GraphFactory，使其能够正确判断所读文件正确性 | 未完成 |
| 05-14 | 16：30-17：30 | 继续上述任务 | 完成 |
| 05-16 | 19：00-21：30 | 学习l并添加og相关内容 | 完成 |
| 05-17 | 16：00-17：30 | 为异常添加部分测试 | 完成 |
| 05-17 | 19：00-21：30 | 为异常添加部分测试 | 完成 |
| 05-18 | 14：00-17：00 | 为异常添加所有测试 | 完成但发现之前编写的测试出现问题 |
| 05-18 | 19：30-21：00 | 修改之前的测试文件 | 完成 |
| 05-19 | 13：30-16：00 | 编写实验报告 | 大部分完成 |
| 05-19 | 18：00-21：00 | 做debug部分内容，并添加测试用例 | 完成 |
| 05-20 | 10：00-11：30 | 发现之前的程序不能通过新添加测试，于是修改 | 完成 |
| 05-20 | 14：00-17：00 | 写实验报告，过程中发现错误，于是同时进行修改 | 完成 |
| 05-20 | 19：00-21：30 | 发现log的查询部分忘做了，于是补做 | 完成 |
|  |  |  |  |

# 实验过程中遇到的困难与解决途径

1. 在修改和添加新内容后，程序不能通过回归测试

解决：检查原来编写的测试，发现对原有合法的测试抛出异常，于是修改原有测试使其符合新的要求。

1. 没有弄清各类型点、边之间的继承关系，checkrep不能准确确定输入信息的正确性。

解决：查阅实验3手册，弄清楚各点、边的具体要求和主次关系，例如现在vertex类中确定label不为空，再在person类型中确定性别和年龄。

3部分点类型不能判断是否重复和label重复

解决：word点类型只有label信息，判断类型只能比较label是否相等，所以只能将label相等的点看做点也相同。

4.socialnetwork中加入权值为1.0的边后会使以前所以边的权值为0，不能保证checkrep的正确性

解决：认为实验要求不准确，于是把边的合法权值设为[0,1]。在加入权值为1.0的边后，其他边的权值改为0.0

5.完成实验过程中因为粗心遇到许多小问题

解决：例如未注意方法选错，未注意{}的使用导致测试出错，花费许多时间排除这样的错误实在觉得不值得，以后一定细心对待。

# 实验过程中收获的经验、教训、感想

1. 经过本次实验，再次加强了对正则表达式的掌握，了解了其在确定程序正确性和健壮性中的重要作用。
2. 意识到checkrep和assertion的意义：完善的checkrep和assertion能够从创建点和边过程中发现错误，于是避免了运行graph产生错误。
3. 认识到修改程序可能会与之前的测试产生矛盾，所有修改时应充分考虑之前的逻辑和情况。
4. 使用log可以记录程序在内部自行解决的错误和所有操作，比直接抛出相比更能发现潜在的问题。
5. 在本次实验中，意识到健壮性和正确性相比，正确性在所有输入数据均合法的情况下，确定进行正确操作可以使程序的功能达到预期；健壮性则需要考虑输入数据是否合法 ，是否进行非法操作等情况。实现正确性主要靠充分考虑给定的要求，健壮性则需要广泛的考虑意料之外的情况。
6. 为了应对1%可能出现的错误或异常，需要增加很多行的代码，我认为并不划算，很多错误情况较为特殊，也行运行很多次都不会遇上。所有使程序发生错误的概率很小，话费时间解决得不偿失。而且我认为对此类错误产生的原因可事先申明以避免其产生。
7. 通过编写不同类型异常，checkrep、assertion、log的使用不但能发现跟多的错误，而且能较为准确定位错误产生的位置用以分析产生错误的原因。
8. 我认为本次实验在给定的时间下，checkrep、异常的编写和测试需要考虑的情况过多导致时间略紧，所以很多新内容不能充分学习，许多情况不能考虑充分。于是本次实验的质量有所降低。