

**2025年春季学期  
计算学部《软件工程》课程**

**Lab 1实验报告**

|  |  |  |
| --- | --- | --- |
| **姓名** | **班级/学号** | **联系方式** |
|  |  |  |

**目 录**

[1 实验要求 1](#_Toc166175466)

[2 待求解问题描述 1](#_Toc166175467)

[3 算法与数据结构设计 1](#_Toc166175468)

[3.1 设计思路与算法流程图 1](#_Toc166175469)

[3.2 数据结构设计 1](#_Toc166175470)

[3.3 算法时间复杂度分析 1](#_Toc166175471)

[4 实验与测试 1](#_Toc166175472)

[4.1 读取文本文件并展示有向图 2](#_Toc166175473)

[4.2 查询桥接词 2](#_Toc166175474)

[4.3 根据桥接词生成新文本 2](#_Toc166175475)

[4.4 计算最短路径 3](#_Toc166175476)

[4.5 随机游走 3](#_Toc166175477)

[5 编程语言与开发环境 3](#_Toc166175478)

[6 结对编程 3](#_Toc166175479)

[6.1 分组依据 3](#_Toc166175480)

[6.2 角色切换与任务分工 3](#_Toc166175481)

[6.3 工作照片 4](#_Toc166175482)

[6.4 工作日志 4](#_Toc166175483)

[7 Git操作过程 4](#_Toc166175484)

[7.1 实验场景(1)：仓库创建与提交 4](#_Toc166175485)

[7.2 实验场景(2)：分支管理 4](#_Toc166175486)

[8 在IDE中使用Git Plugin 4](#_Toc166175487)

[9 小结 4](#_Toc166175488)

[文档全部完成之后，请在上述区域点击右键，选择“更新域”，在打开的对话框中选择“更新整个目录”]

# 实验要求

简要复述实验手册中要求达到的实验目标与要求。

# 待求解问题描述

针对本实验的从文本生成图并在图上进行运算的要求，给出详细描述。不能照搬实验手册上的说明，应增加自己的理解。

需包含：输入数据、输出数据、约束条件等。

# 算法与数据结构设计

## 设计思路与算法流程图

给出文字描述的算法设计思路。

使用大一高级语言编程课里所学知识，用流程图的方式描述算法设计思路，让读者清晰理解。如果需要，可针对每一个模块分别绘制流程（根据文本生成图、展示图、查询桥接词、根据桥接词生成新文本、计算最短路径、计算PR值、随机游走）。

如采用大模型辅助，给出各部分对应的提示词。并描述对大模型产生结果进行的分析和评价，发现的问题，如何进行的修改。

## 数据结构设计

使用大二期间数据结构的相关知识，给出算法中所用的每个数据结构的定义。至少需定义本题中要求生成的有向图的数据结构。

如采用大模型辅助，给出各部分对应的提示词。并描述对大模型产生结果进行的分析和评价，发现的问题，如何进行的修改。

## 算法时间复杂度分析

使用大二期间算法与数据结构课程的相关知识，简要分析你设计的算法的时间复杂度。

可针对每一个模块分别分析（根据文本生成图、展示图、查询桥接词、根据桥接词生成新文本、计算最短路径、计算PR值、随机游走）。

如采用大模型辅助，给出各部分对应的提示词。并描述对大模型产生结果进行的分析和评价，发现的问题，如何进行的修改。

## 算法代码的生成

给出采用大模型辅助生成代码的提示词，并描述对大模型产生代码进行的分析和评价，发现的问题，如何进行的修改。

# 实验与测试

利用提供的Easy Test.txt文件和Cursed Be The Treasure.txt两个文件进行测试。

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。**下面各项只需要填写针对Easy Test.txt的实验结果。**

## 读取文本文件并展示有向图

文本文件中包含的内容：

期望生成的图（手工计算得到）：

程序实际生成的图：

二者是否一致：

给出实际运行得到结果的界面截图。

## 查询桥接词

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 输入（2个单词） | 期望输出 | 实际输出 | 运行是否正确 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

给出实际运行得到结果的界面截图。

## 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 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

给出实际运行得到结果的界面截图。

## 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 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

给出实际运行得到结果的界面截图。

## 计算PageRank值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 单词 | 期望输出 | 实际输出 | 运行是否正确 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

## 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 |  |  |
| 2 |  |  |
| 3 |  |  |

给出实际运行得到结果的界面截图。

# 编程语言与开发环境

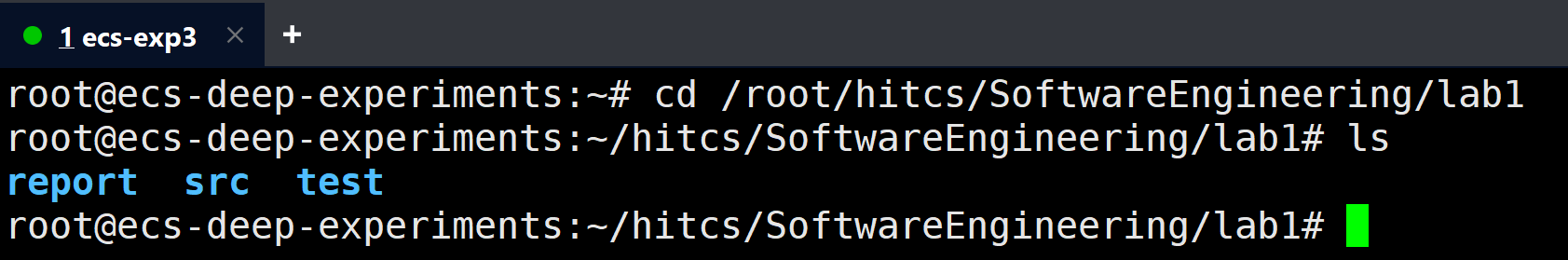
Java DK版本、IDE版本、其他需要额外说明的开发配置；

采用的大模型。

# Git操作过程

## 实验场景(1)：仓库创建与提交

进入项目目录：



**R1：本地初始化一个git仓库，将自己在Lab1中所创建项目的全部源**

**文件加入进去，纳入git管理；**

文本

AI 生成的内容可能不正确。

图形用户界面, 文本

AI 生成的内容可能不正确。

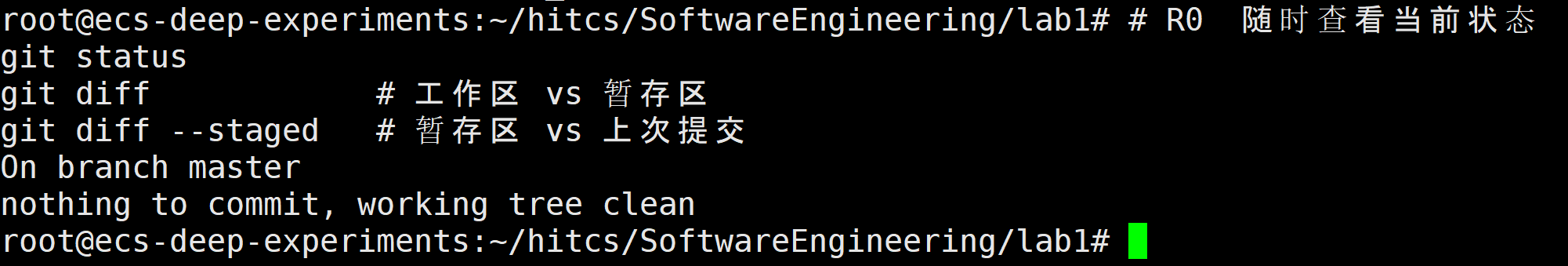
第一次提交：§ **R2：提交；**

文本

AI 生成的内容可能不正确。

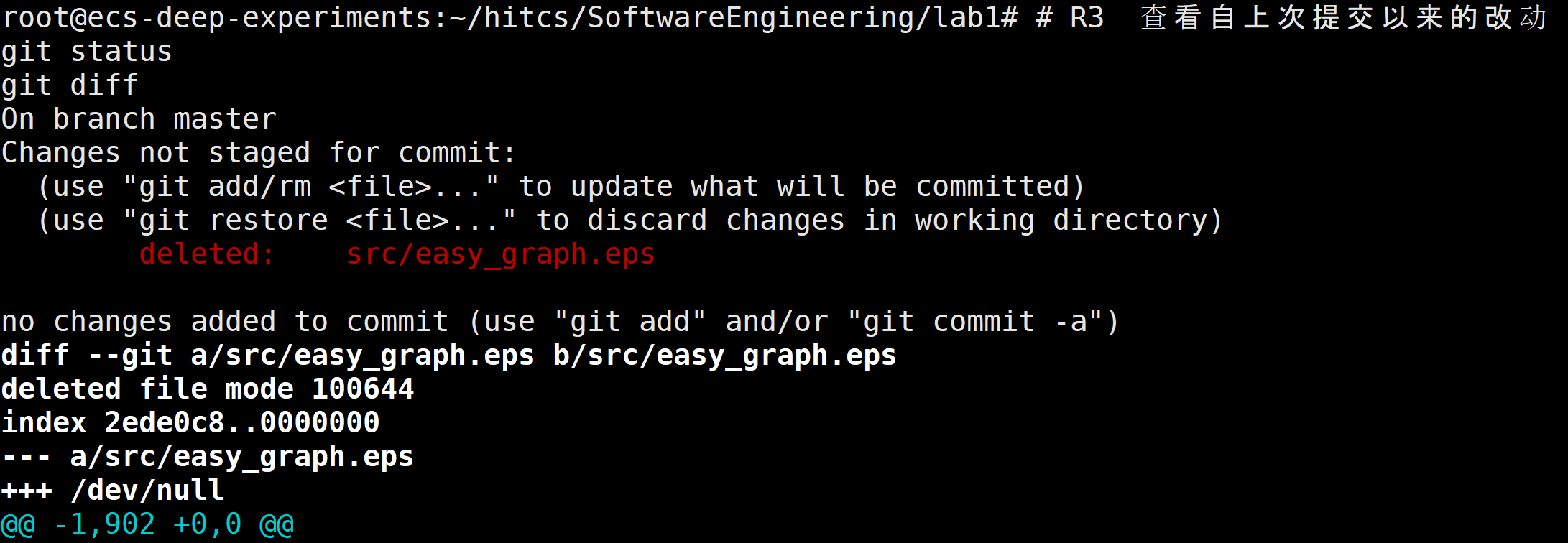
§ **R0：在进行每次git操作之前，随时查看工作区、暂存区、git仓库的**

**状态，确认项目里的各文件当前处于什么状态；**



提交后，第一次修改，删除了一个eps文件：**手工对提交的部分文件进行修改；**

§ **R3：查看上次提交之后都有哪些文件修改、具体修改内容是什么；**



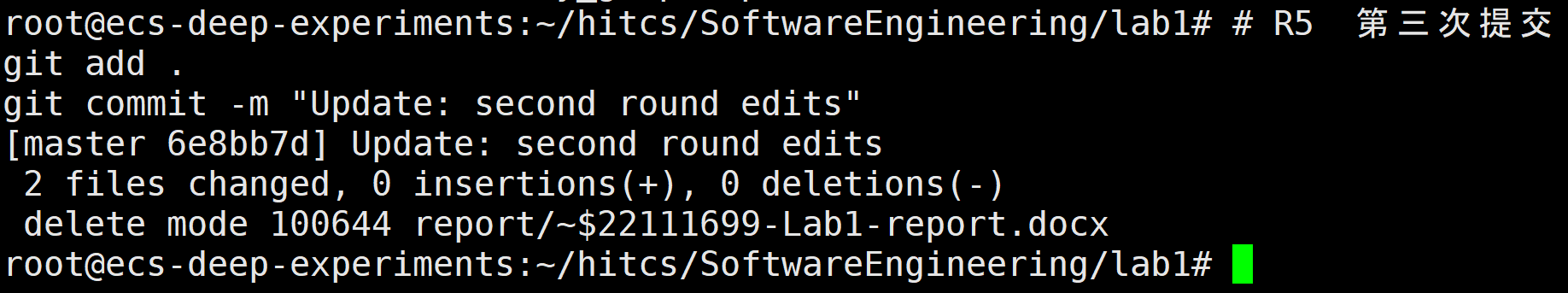
第二次提交：**R4：重新提交；**

文本

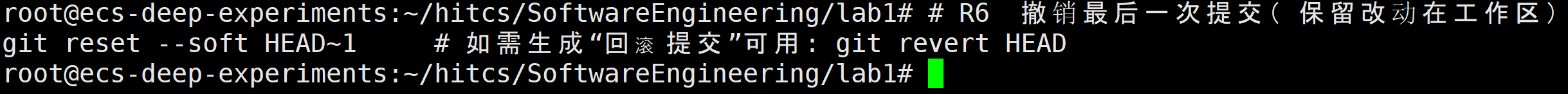
AI 生成的内容可能不正确。

第二次修改：删除了一个临时文件，修改了一个.docx文件。**再次对部分文件进行修改；**

第三次提交：§ **R5：重新提交**



§**R6：把最后一次提交撤销；**

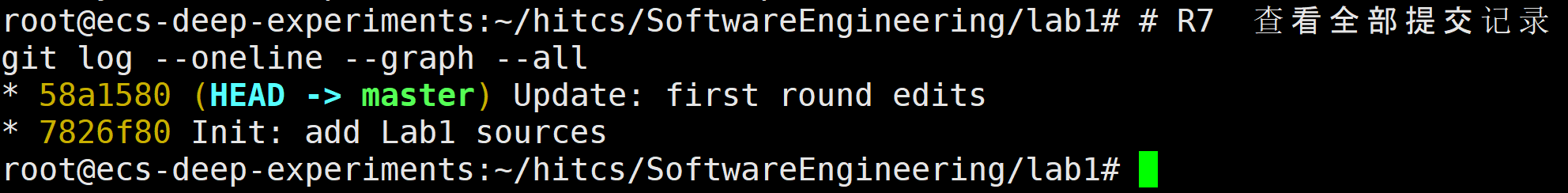


查看状态：§ **R0：在进行每次git操作之前，随时查看工作区、暂存区、git仓库的**

**状态，确认项目里的各文件当前处于什么状态；**



§**R7：查询该项目的全部提交记录；**

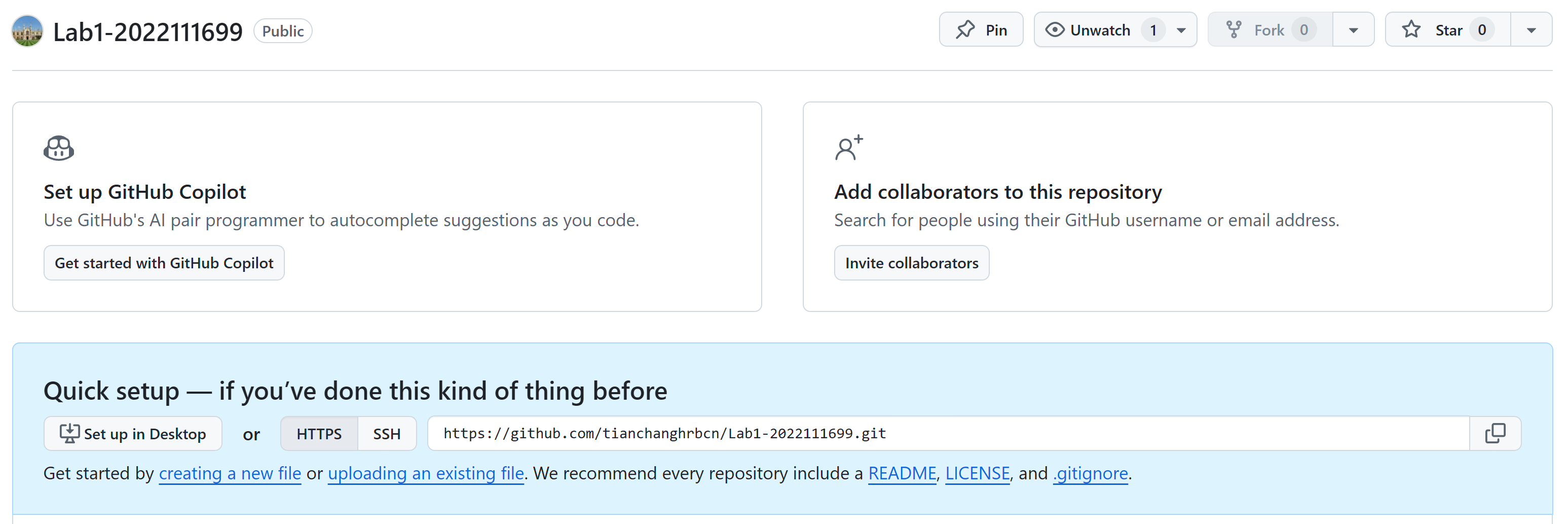


§**R8：在GitHub上创建名为“Lab1-学号”的仓库，并在本地仓库建立相**

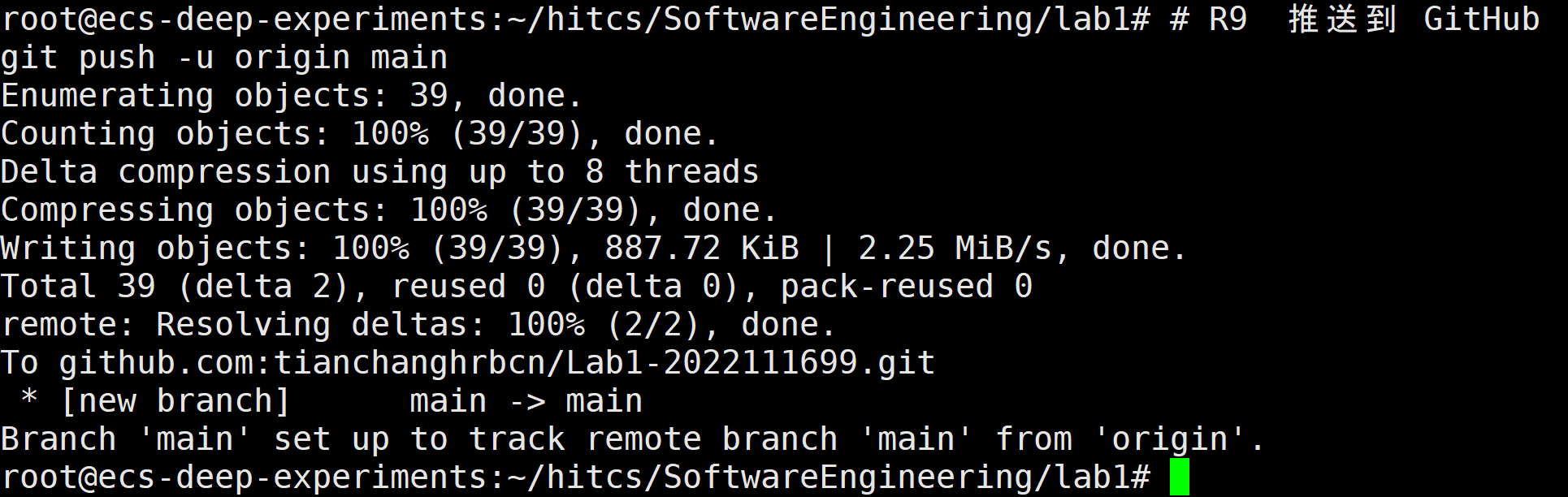
**应的远程仓库；**

文本

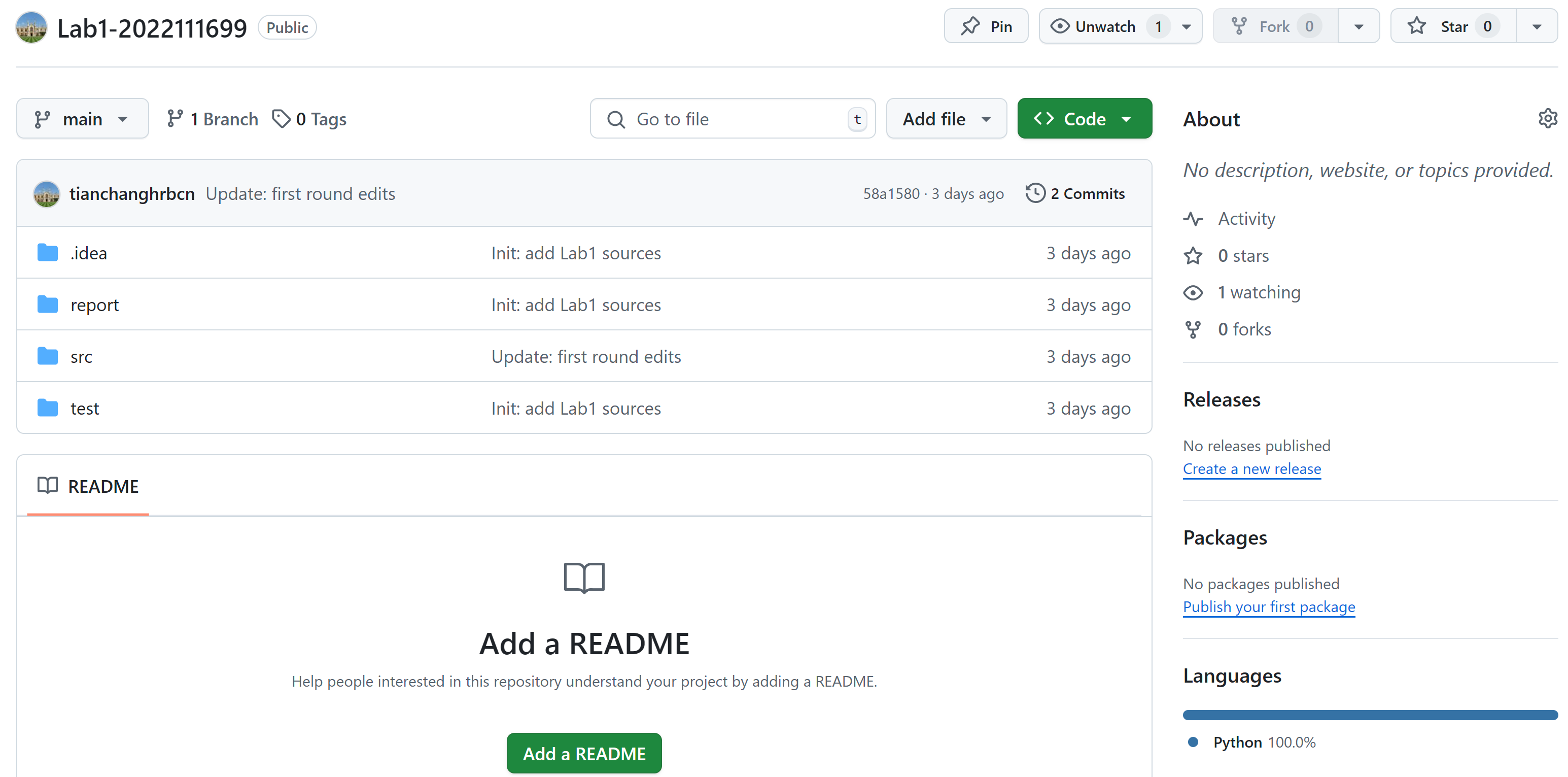
AI 生成的内容可能不正确。



§**R9：将之前各步骤得到的本地仓库全部内容推送到GitHub的仓库中；**

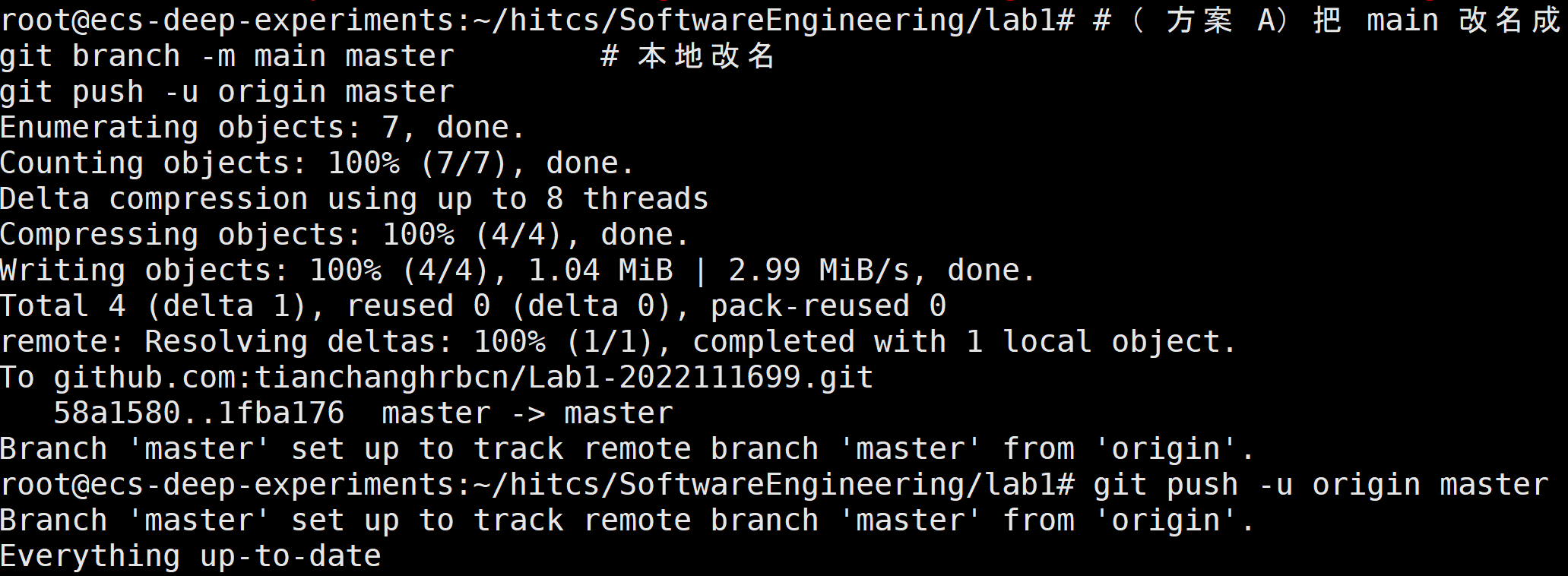
****

**仓库界面截图：**

****

## 实验场景(2)：分支管理

从场景1继续，由于我的github默认分支是main，所以先在本地和远程改分支名：



图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

接下来按照步骤进行：

**§ R1：获得本地Lab1仓库的全部分支，切换至分支master；**

文本

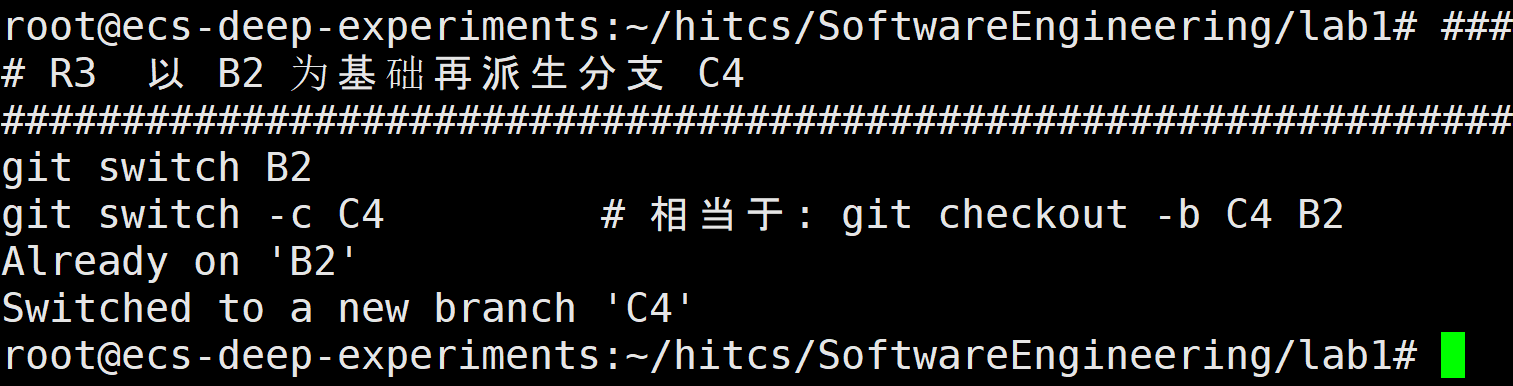
AI 生成的内容可能不正确。

§ **R2：在master基础上建立两个分支B1、B2；**

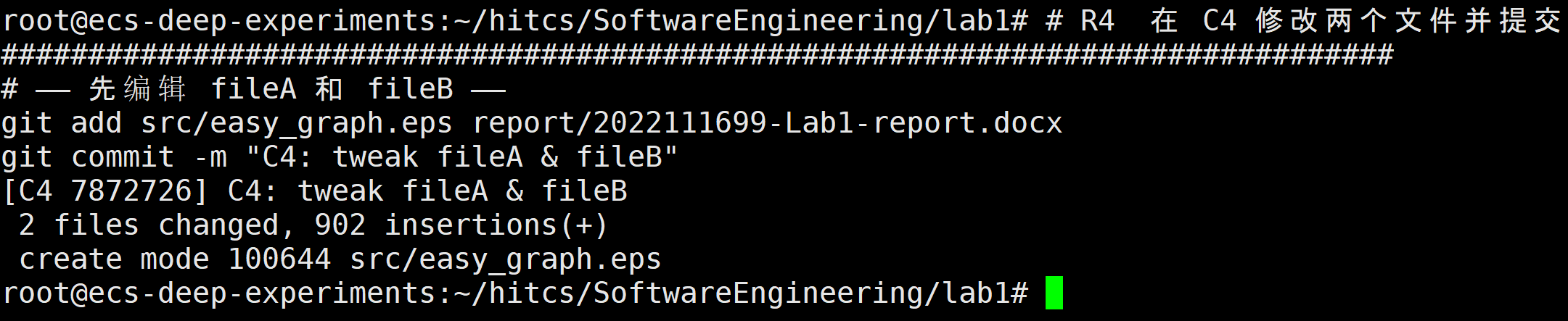
文本

AI 生成的内容可能不正确。

§ **R3：在B2分支基础上创建一个新分支C4；**



§ **R4：在C4上，对2个文件进行修改并提交；**



§ **R5：在B1分支上对同样的2个文件做不同修改并提交；**

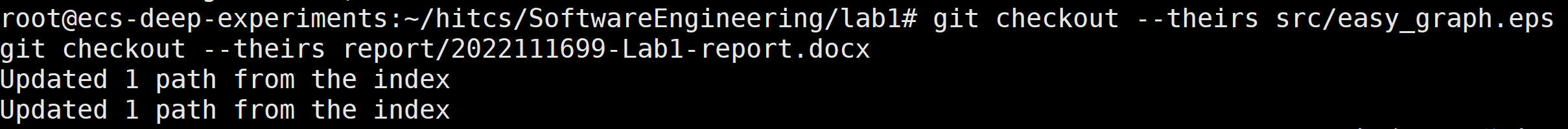


§ **R6：将C4合并到B1分支，若有冲突，手工消解；**

文本

AI 生成的内容可能不正确。

修改冲突：

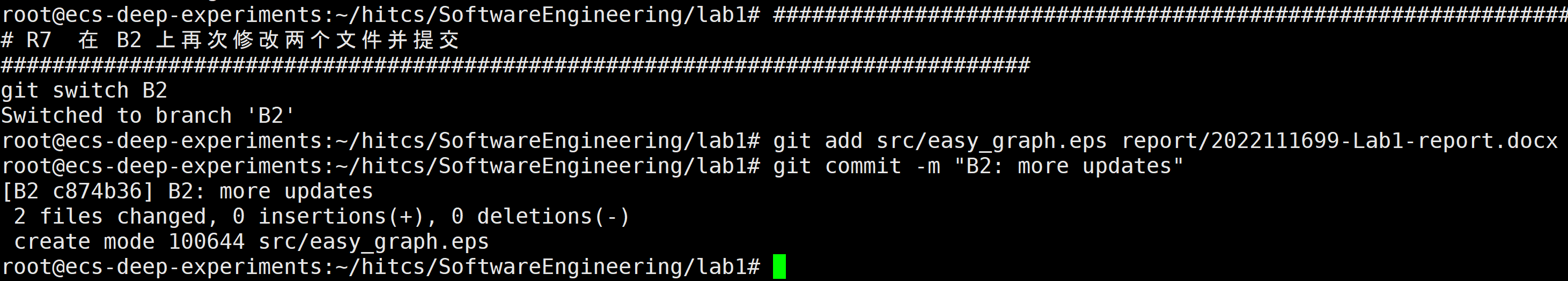


再次提交：

文本

AI 生成的内容可能不正确。

§ **R7：在B2分支上对2个文件做修改并提交；**



§ **R8：查看目前哪些分支已经合并、哪些分支尚未合并；**

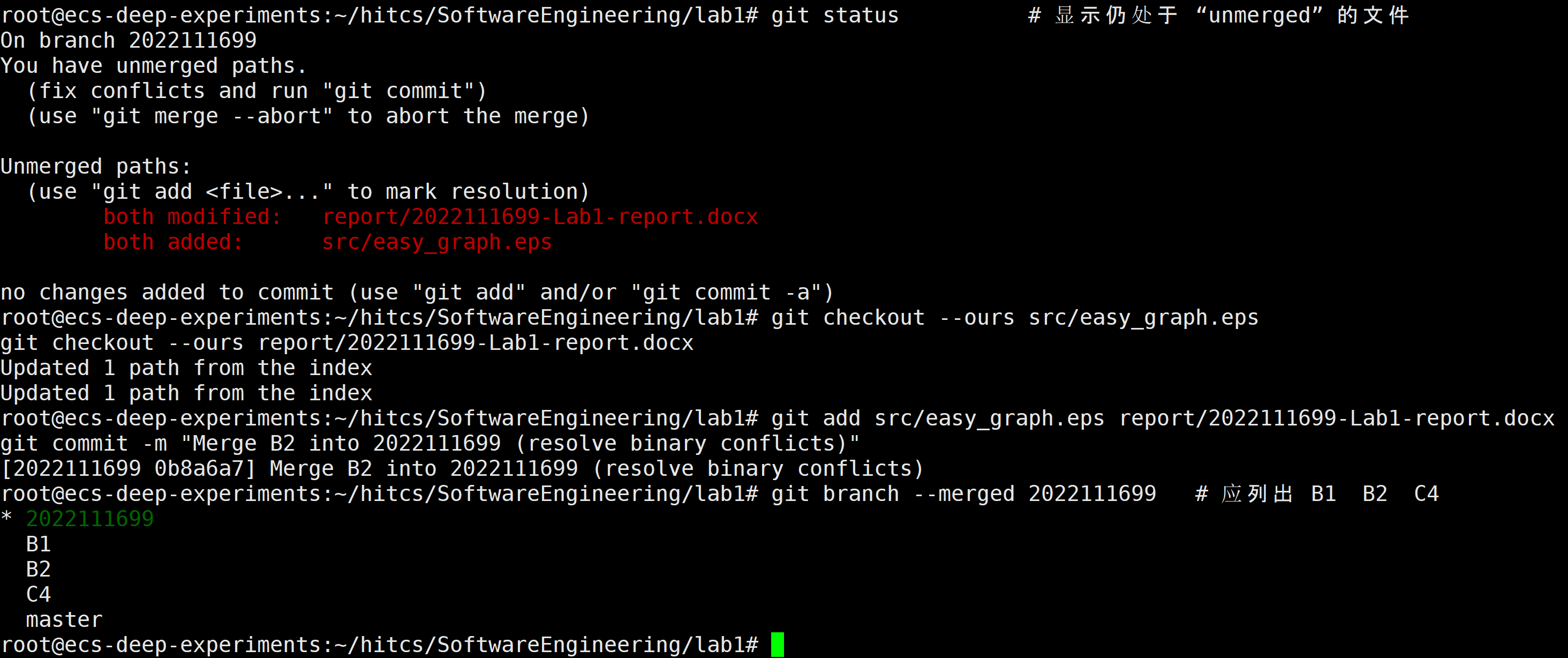
文本

AI 生成的内容可能不正确。

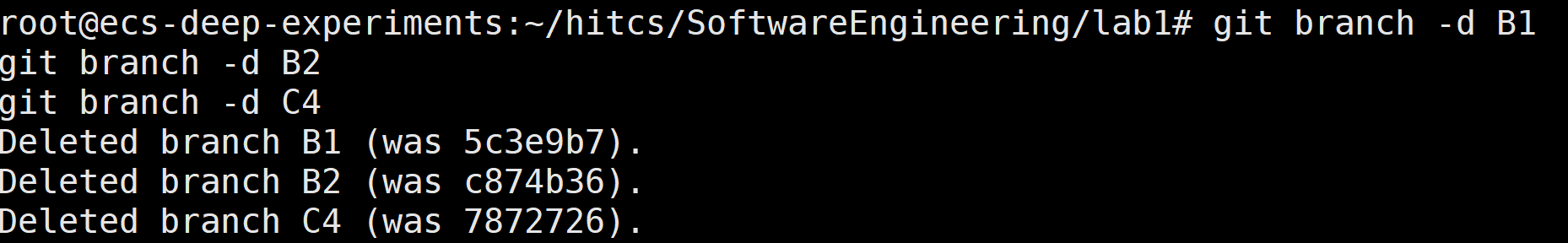
§ **R9：将已经合并的分支删除，将尚未合并的分支合并到一个新分支上，**

**分支名字为你的学号；**

**首先检查分支状态，处理每个冲突的文件，确定需要哪个版本：**



删除多余的分支：

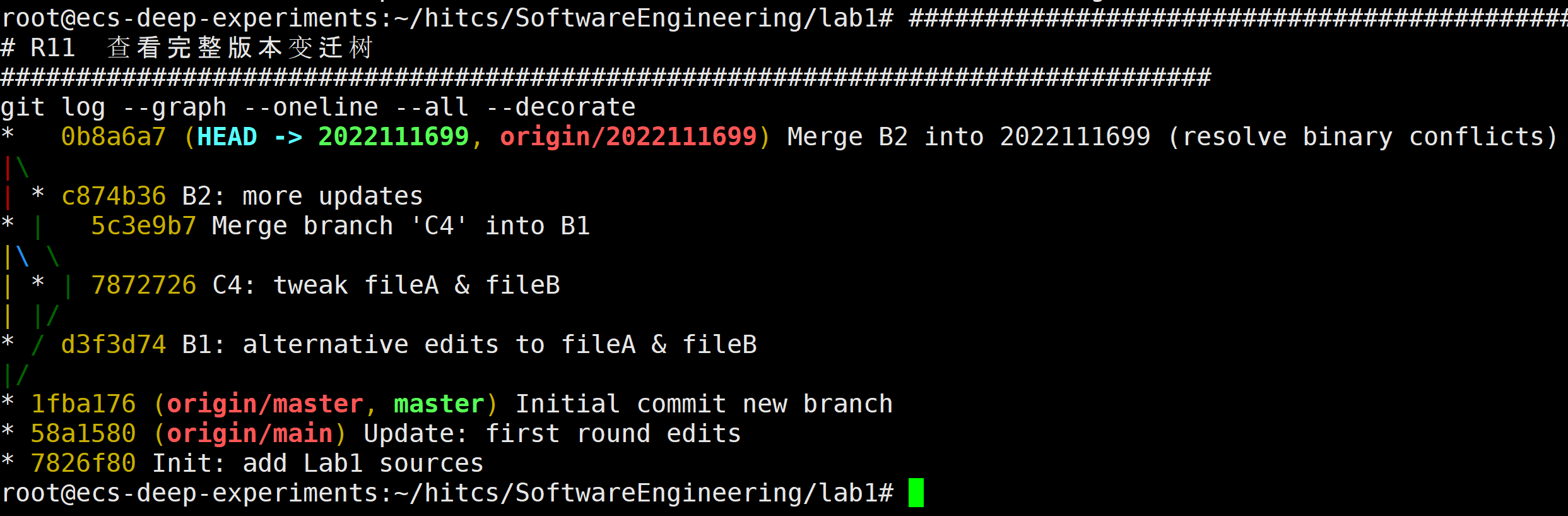


§ **R10：将本地以你的学号命名的分支推送到GitHub上自己的仓库内；**

文本

AI 生成的内容可能不正确。

§ **R11：查看完整的版本变迁树；**



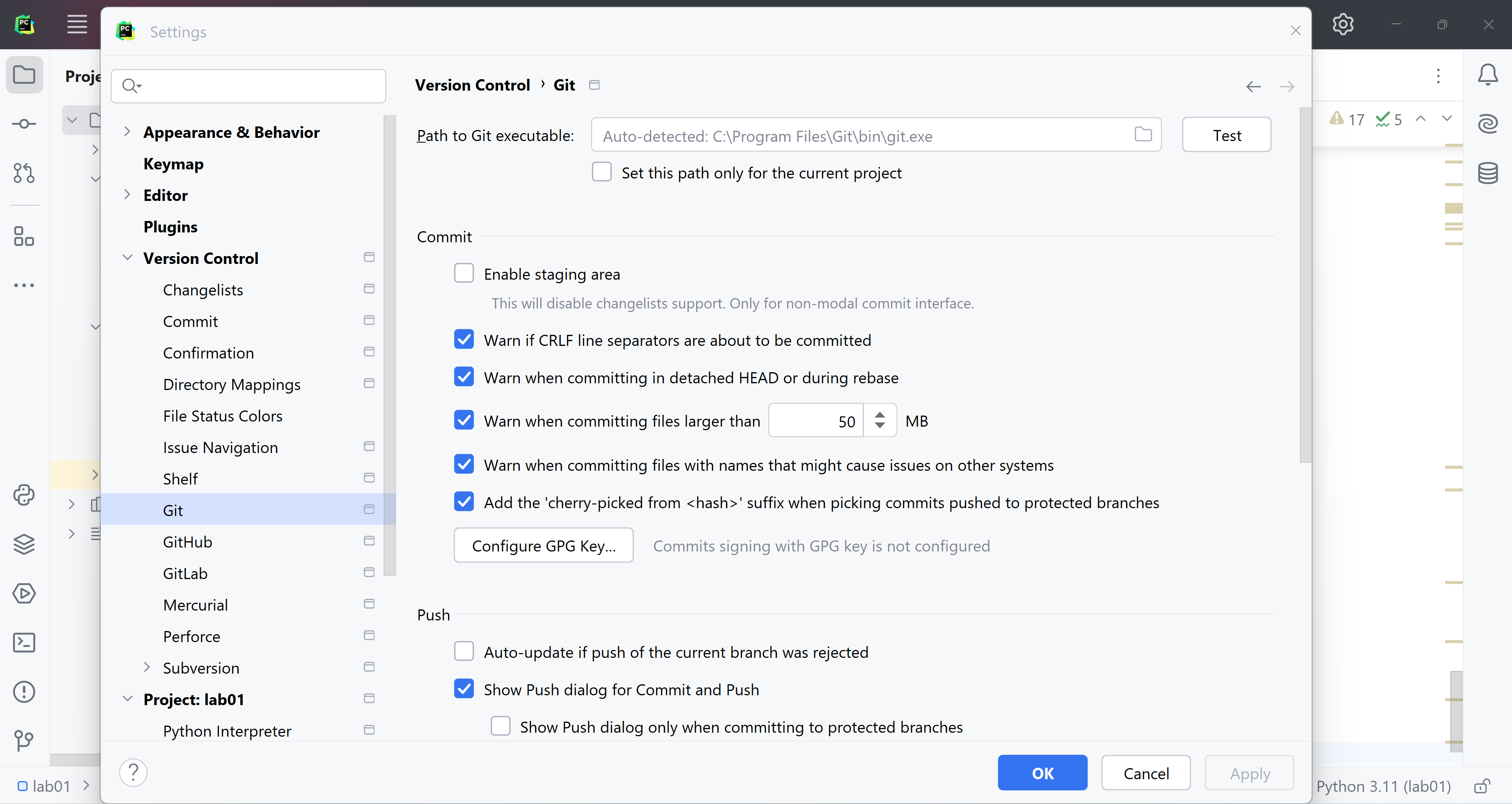
§ **R12：在Github上以web页面的方式查看你的Lab1仓库的当前状态。**

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

# 在IDE中使用Git Plugin

**在Pycharm 2024中打开git设置，并点击OK：**

****

**Ctrl+K打开commit界面：**

**图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。**

**可以发现有3个文件added，2个文件modified，点击Commit完成提交，点击Commit and Push完成提交并推送。**

# 小结

**对比之前设计和编写程序的方式，总结利用大模型辅助后，带来的优势与劣势。**