* + 计算机专业的本次读书实践周的实施方案

拟：1. git与代码版本管理、学习笔记（必做）；2. 读书笔记（选做）

具体任务：

1、寻找合适的资料进行学习，在本地安装配置git，创建本地仓库；

2、注册远程仓库账号（GitHub、Gitee……），并创建至少一个公开的远程仓库。

2.1 选择可以公开的本地代码推送到远程仓库；

2.2 至少提交三次代码，观察每次提交的变更。

……

1. 建一个公开仓库，写本次实践任务的学习笔记。笔记包括：学习资料来源、链接等；实践流程；遇到的困难及解决方法；心得等部分。欢迎推荐书籍给老师。

成果收集：

各班拟以excel形式收集远程git账号地址，并检查公开仓库、学习笔记、读书笔记可访问性，体现在excel中；

* 大二同学按照计算机网络课程要求完成项目，并更新学习笔记到git。
* 大三的同学按照操作系统课程要求完成项目，并更新学习笔记到git。
* 大四同学将自己的毕设代码更新到git，并分享给导师。如果不适合公开，可以设置为私有仓库。如果不适合上网，请在读书笔记中注明。
* 已经使用过git的同学，可以在上学期的书单中继续选择书目进行阅读，并在git上更新读书笔记。书单地址：<https://docs.qq.com/doc/DWFNlbE5ld3FQVE9v> 欢迎推荐合适的书籍。

本次实践目的：方便评估学生到毕业时整个大学四年的代码工作量。

前期学习资料来源：

* 1. DeepSeek
  2. Git常用命令大全：https://www.runoob.com/note/56524

实践流程：

1. 下载Git。从官网安装Git并按照指引完成勾选。
2. 已注册Github账号，创建一个个人仓库。
3. 本地git准备：

（1）git –version 查看是否安装成功

（2）git config --global user.name "Your Name"

git config --global user.email [email@example.com](mailto:email@example.com)

在Git Bash中使用这两个命令创建自己的连接相关配置

Git config –list 进行查看

（3）使用cd /d/ 命令跳转到任务目录

并使用git init命令初始化一个空的.git目录

（4）touch README.md

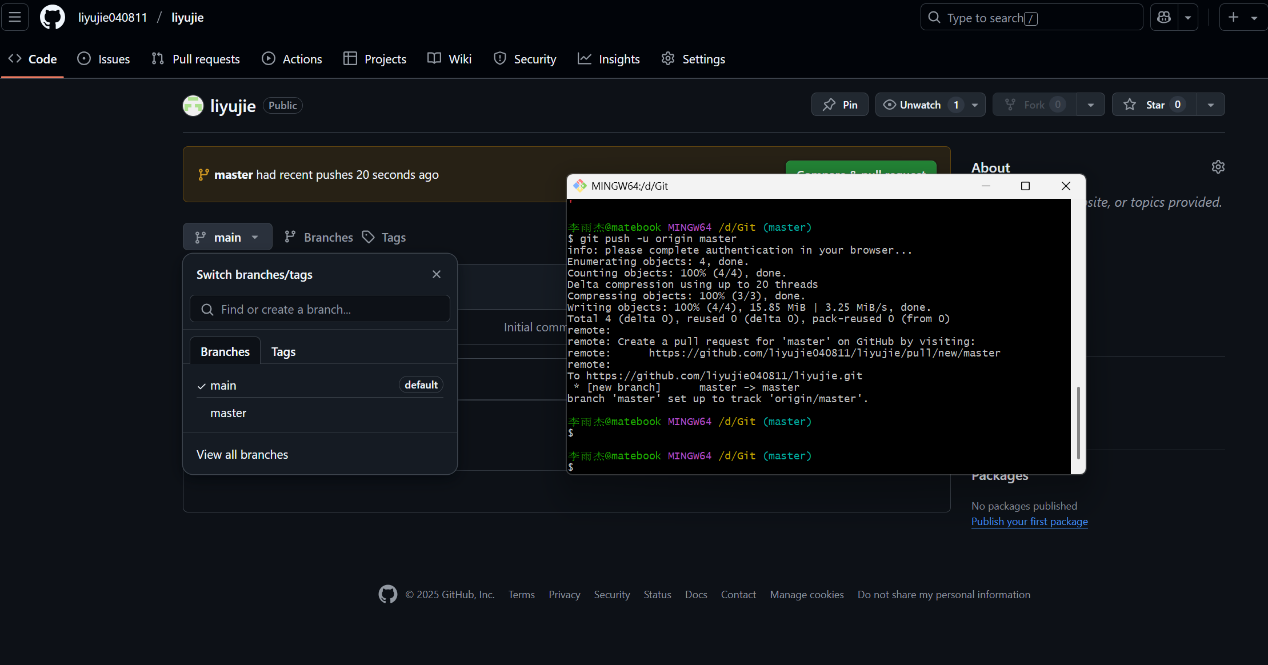
git add README.md

git commit -m "first"添加文件并提交

1. GitHub准备：

创建一个空的公共仓库，默认分支为main

1. 链接远程仓库与本地仓库：
2. git remote add origin <https://github.com/liyujie040811/liyujie.git>命令关联本地与远程仓库
3. 接着使用git push -u origin master推送代码，推送后GitHub变更



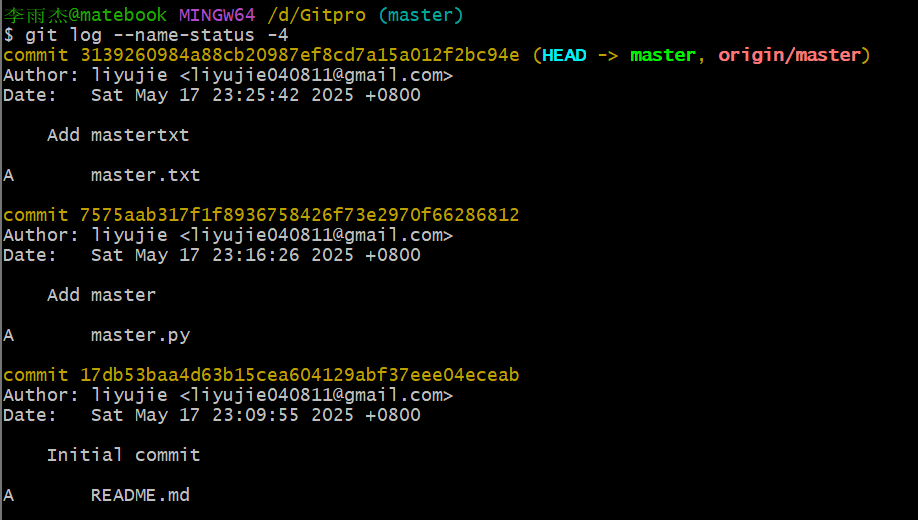
git push -u origin master（这里还要看bash里面路径之后的标记是main

还是master，是版本问题，也是项目仓库的分支。如果本地分支和远程分支冲突，可以使用：git branch -m master main # 将本地分支重命名为 main

1. 提交三次变更：

第一次提交：修改README.md，添加项目描述。git add README.mdgit commit -m "Add project description"git push第二次提交：新增文件main.py，添加代码。touch main.pygit add main.pygit commit -m "Add main script"git push第三次提交：修改代码并删除无用文件。git add . && git commit -m "Optimize code and remove temp files"git push

使用$ git log --name-status -3观察三次提交的变更



**注意**：**1.**不要将操作路径放在Git的安装路径下，会导致 git add 尝试跟踪 Git 的系统文件（如 git-bash.exe、mingw64/ 等）

解决方案：# 退出当前 Git 安装目录

cd /d/

# 创建新项目目录（例如 my-project）

mkdir my-project && cd my-project

# 将 README.md 移动到新目录（原路径为 D:/Git/README.md）

mv /d/Git/README.md ./

# 初始化新仓库

git init

touch ,gitignore添加.gitignore文件过滤 Git 安装目录的系统文件，避免后续出现 Untracked files 警告

git add .gitignore

git commit -m "Add .gitignore"提交

然后重新关联GitHub并推送

$ git push -u origin master

fatal: unable to access 'https://github.com/liyujie040811/liyujie.git/': Failed to connect to github.com port 443 after 21073 ms: Could not connect to server

推送代码时因无法连接 GitHub 的 443 端口导致失败，直接推送不安全，可通过rm -rf .git

删除配置进行重新关联

**2.**若对某个文件夹无权限，可使用git config --global --add safe.directory /将目录添加到信任列表

**3.**对于本地与远程仓库的关联，我是通过git remote add origin <https://github.com/liyujie040811/liyujie.git>命令在浏览器弹出一个弹窗使用邮箱进行关联。但是我更加推荐生成SSH key：ssh-keygen -t ed25519 -C [xxxx@xxx.com](mailto:xxxx@xxx.com)。输入这个命令后会弹出“Enter file in which to save the key”，你可以选择给你的密钥文件重新命名，也可以直接回车使用默认的命名（id\_ed25519）。之后会让我们给密钥添加密码，当然也可以不输入。接着前往C>user>username>.ssh文件夹里面找刚刚生成的密钥文件，复制公钥文件内容，私钥不要泄露。公钥就是带.pub结尾的文件，将.pub文件打开并将里面的内容全部复制。打开Github并找到settings，在左侧列表选择SSH and GPG keys，添加新的ssh key。给这个ssh连接命名，并将刚才复制的公钥全文复制到下方文本框内点击创建。创建后会发现shh的钥匙图标是灰色的，是因为本地还没有和github产生链接。找到自己想要管理的代码文件夹，并在git bash里面使用cd命令将文件路径指向这个文件夹。

**感触：**git对我来说是一个新的管理代码的工具，由于之前未了解过，在过程中也经历了许多困难，多亏了现在的大模型与编程社区，让解决问题变得简单。学习Git的过程就是不断试错和探索，并去理解的过程，上头的同时又有攻克难关的成就感。