

Git 手动合并修改指南

首先进入您的代码仓库目录：

```
cd 你的目录名
```

强烈推荐：使用 VSCode 打开项目，便于后续查看和解决冲突

```
code .
```

步骤 0：stash

由于您大概更改了 `bits.c`，使用 `git stash`` 保存当前更改

```
git stash
```

步骤 1：添加远程仓库

```
git remote add template https://github.com/RUCICS/DataLab-2025Fall.git
```

⚠ 注意：此命令只需执行一次

步骤 2：获取远程仓库更新

```
git fetch template main
```

步骤 3：合并远程仓库更新

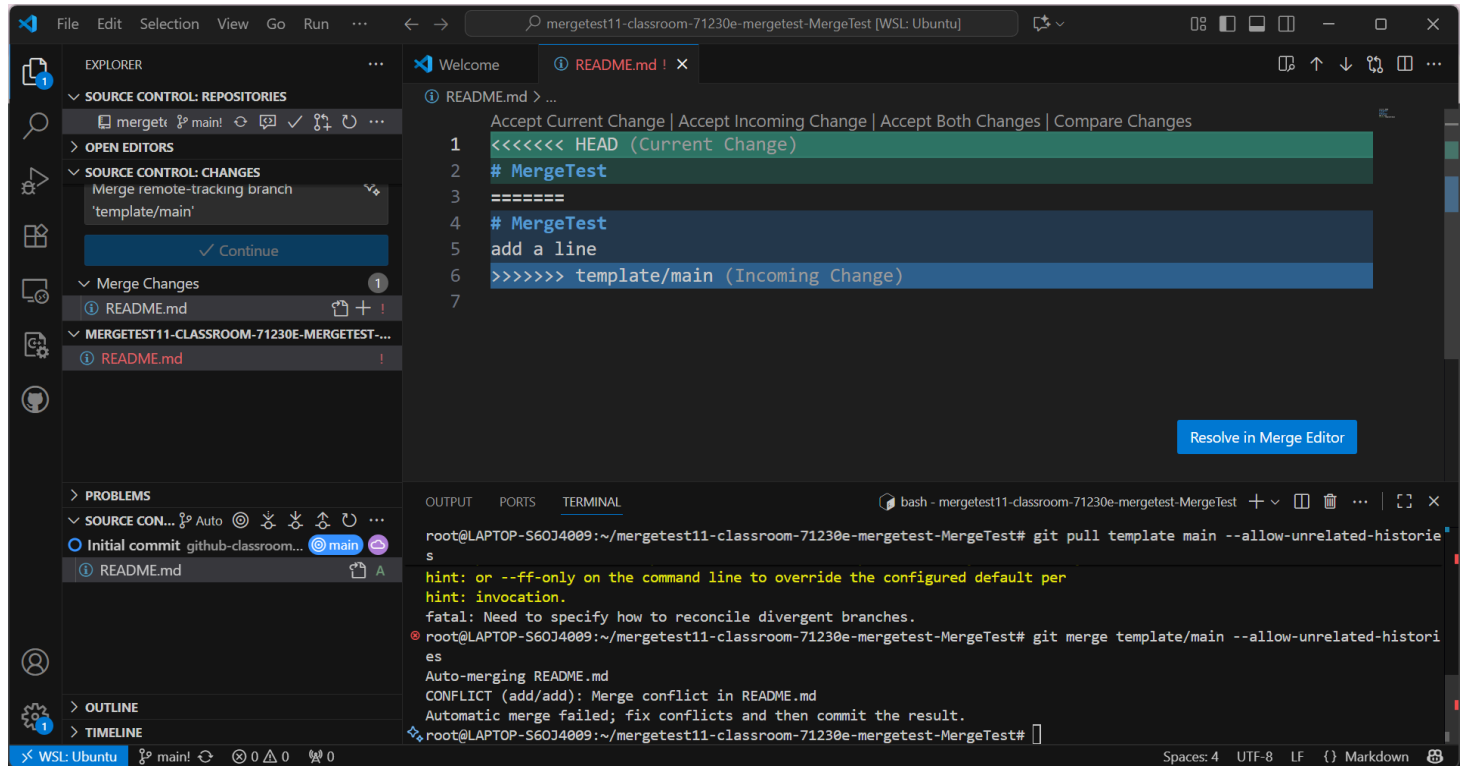
使用 rebase

```
git rebase template/main
```

步骤 4：解决冲突

情况一：使用 VSCode 解决冲突

如果您已经运行了 `code .`，可以直接在 VSCode 中解决冲突：



1. 查看冲突文件

- 在左侧面板的 `SOURCE CONTROL: CHANGES` 中查看所有冲突文件

2. 解决冲突

- 点击打开冲突文件
- 在冲突区域选择 **Accept Incoming Change**

 **说明：**以后根据实际情况，您可以选择：

- Accept Current Change - 接受当前分支的修改
- Accept Incoming Change - 接受远程分支的修改
- Accept Both Changes - 接受双方的修改

当前建议选择 `Accept Incoming Change`

3. 添加到暂存区

- 点击文件右侧的 "+" 号，将修改添加到暂存区



情况二：使用命令行编辑器解决冲突

如果没有使用 VSCode，需要通过命令行编辑器（如 vim 或 nano）解决：

如果是 Vim：

- 按 `Esc` 键退出插入模式
- 输入 `:wq` 保存并退出

如果是 Nano：

- 按 `Ctrl + X` 退出
- 按 `y` 确认保存
- 按 `Enter` 确认文件名

解决完冲突后...继续rebase

1. 检查确认文件无误后，暂存所有更改

```
git add .
```

2. 将这些更改追加到当前正在编辑的提交中，并且不修改提交信息

```
git commit --amend --no-edit
```

3. 继续 rebase 过程

```
git rebase --continue
```