Git and LaTeXWorksheet

Justin Li

justinnhli@oxy.edu

Occidental College

1 Instructions

This worksheet is due March 1, 2025 at midnight, to be submitted as a GitHub repository URL to Canvas. The repository should contain all files requires to compile this worksheet with your answers. You should only change this `document.tex` file and the `references.bib` file; do not change any other file in this starting repository. You should not use any additional packages, and are not allowed to use the `\usepackage{}` command. Additionally, the output should be formatted correctly: your answers should be appropriately nested under the questions, command-line commands should be in monospace, and images should be positioned appropriately.

2 Git Questions

2.1 General questions

1. What is a version control system? Why are they useful?
   A version control system tracks changes to files over time and allows for collaboration, access to change history, and the ability to recover previous versions. It is useful for managing a codebase efficiently, especially when collaborating with others.
2. What is the difference between git and GitHub?
   Git is a tool that is already on my computer. GitHub is a cloud-based platform that allows people to host Git repositories and provides tools for collaboration, such as making pull requests or tracking issues.
3. What is a repository?
   A repository is a project directory managed by Git that tracks file changes and stores commits.
4. What is a commit?
   It is a snapshot or a version of changes in the repository at a specific point in time. It has a unique hash for identification.
5. What is the commit graph?
   A commit graph is a visual representation of commits and their relationships to each other. It is displayed as nodes connected by arrows that shows the history of a project.
6. What is your preferred local git client (eg., command line, GitHub Desktop, GitKraken, etc.)?
   I usually use the command line but have used GitHub Desktop as well.

2.2 Local Usage

1. What is the difference between adding a file to the staging area and committing a file?
   Adding a file to the staging area prepares it for inclusion in the next commit. Committing records the staged changes permanently in the repository's history.
2. What is a commit message, and why is it important for them to be meaningful?
   A commit message is just a description of the changes that were made. It is important for them to be meaningful to help you, your collaborators, and the viewers of the repository understand the purpose of each commit. It can also help when you want to revert to a previous version.
3. Starting with an empty repository, what sequence of commands/actions would result in the following commit graph? You may give a sequence of `git` commands, or describe (with screenshots) how you would do this in your preferred graphical git interface.

   ```
   A---B---C---D
   ```

   ```
   git init
   echo "A" > file.txt
   git add file.txt
   git commit -m "A"
   echo "B" >> file.txt
   git add file.txt
   git commit -m "B"
   echo "C" >> file.txt
   git add file.txt
   git commit -m "C"
   echo "D" >> file.txt
   git add file.txt
   git commit -m "D"
   ```

4. If you are currently at commit D above, how would you recover code from commit B? What sequence of commands/actions would let you do so? You may give

a sequence of `git` command-line commands, or describe (with screenshots) how you would do this in your preferred graphical git interface. Assume the commit hashes are AAAAAA..., BBBBBB..., etc.
To just look at the code from commit B:

```
git checkout BBBBBB
```

To create a new branch from commit B:

```
git checkout -b new-branch-name
BBBBBB
```

5. Imagine you created a git repository for your project, but only commit your changes once a week on Sundays. You got your code working on Tuesday, but then broke your code on Friday. What can you do to get the working version of your code back?
If no changes were committed or stashed, there's nothing you can do. This is why you should commit often to avoid these types of situations.

2.3 Branching and Merging

1. What is a branch? Why are they useful?
A branch is essentially a pointer to a specific commit that allows you to diverge from the main development code. It's very useful for developing new features and working independently on a large codebase.
2. Starting with an empty repository, what sequence of commands/actions would result in the following commit graph? You may give a sequence of `git` command-line commands, or describe (with screenshots) how you would do this in your preferred graphical git interface.

```
A---B---C---D
     \
      E---F
```

```
git init
echo "A" > file.txt
git add file.txt
git commit -m "A"
echo "B" >> file.txt
git add file.txt
git commit -m "B"
git branch feature
git checkout feature
echo "E" >> file.txt
git add file.txt
git commit -m "E"
echo "F" >> file.txt
git add file.txt
git commit -m "F"
git checkout main
echo "C" >> file.txt
git add file.txt
git commit -m "C"
echo "D" >> file.txt
git add file.txt
git commit -m "D"
```

3. Why is a merge? Why are they useful?
A merge combines changes from different branches into a single branch. They are extremely useful for when multiple developers are working on different features for a project and they will need to combine it for the final product at the end. It's also useful because we don't want to mess with the production code when implementing new features. Until it's fully tested, they should remain separate.
4. Imagine you are currently at commit D above. What sequence of commands/actions would result in the following commit graph? You may give a sequence of `git` commands, or describe (with screenshots) how you would do this in your preferred graphical git interface.

```
A---B---C---D---G
     \         /
      E---F---/
```

```
git checkout main
echo "G" >> file.txt
git add file.txt
git commit -m "G"
git merge feature
```

5. What is a merge conflict? When do they occur?
A merge conflict occurs when there are conflicting changes, such as wanting to merge two branches but the same line of a file is changed in both branches. Another example is when one branch modifies a file but the other branch deletes it. Essentially, Git doesn't know which change to keep so the programmer will have to manually resolve that conflict.
6. Starting with an empty repository, despite a sequence of commands/actions that would result in a merge conflict. Include the exact edits and `git` commands or screenshots of the graphical git interface. Include the output or screenshot that shows the resulting merge conflict.

```
git init
echo "Initial info" > file.txt
git add file.txt
git commit -m "Initial commit"
git branch feature
git checkout feature
echo "Feature 1 change" > file.txt
git add file.txt
git commit -m "Commit on feature"
git checkout main
echo "Main branch change" > file.txt
git add file.txt
git commit -m "Commit on main"
echo "Main branch change" > file.txt
git add file.txt
git commit -m "Commit on main"
git merge feature
```

2.4 Remotes

1. What is a remote?
   This refers to when a repository is hosted outside of the local computer such as on a service like GitHub or Bitbucket.
2. What does pushing and pulling do?
   The pushing operation uploads the commits on the local computer to the remote repository. The pulling operation downloads the changes from the remote repository and merges them into the local branch.
3. Imagine you created a git repository for your project on your laptop and commit regularly, but only push your code to GitHub once a week on Sundays. Your laptop caught on fire on Friday. What can you do to get your code back?
   Unfortunately, you can only get the code from last Sunday back by cloning it from GitHub. The commits on the local computer will be gone.

3 LaTeX

Find a source of each of the following types and add it to `references.bib`, with the appropriate data. Your sources do not have to relate to your project. Looking at Overleaf (2025) [7] and Wikipedia (2025) [10] may be helpful,

- a journal article
- a conference article
- a PhD or Master's thesis
- an article in an edited popular media venue (newspaper, magazine, etc.)
- a book
- a chapter of a book
- a YouTube video
- a piece of technical documentation (e.g., a programming language reference, and API documentation, etc.)

Additionally, in you own words, explain the difference between `\cite{}` and `\textcite{}`. When should they each be used? Demonstrate your answers by using one of them with each of your references from above.

`\cite{}` generates an in-text citation enclosed in parentheses. It is not a grammatical part of the sentence. It is also more suitable for general references or when listing multiple sources.

Example:

- TETRIS is a scalable and efficient neural network acceleration framework ([4]).

`\textcite{}` integrates the citation into the text as part of the sentence. It automatically formats the author's name within the sentence along with the year in parentheses. This is useful when you want to emphasize the author or make them the subject of the sentence.

Examples:

- Al-Ansi et al. (2023) [1] analyzed recent developments in augmented and virtual reality for educational applications.

- Gao et al. (2017) [4] introduced TETRIS, a scalable and efficient neural network acceleration framework.

- Kuszmaul (2023) [6] explored novel perspectives and hidden surprises in randomized data structures.

- Boulton (2025) [2] examined whether synthetic data can help scale AI's data limitations.

- Pruksachatkun, Mcateer, and Majumdar (2023) [8] provided guidelines for building consistent, transparent, and fair AI pipelines.

- Hiraga et al. (2024) [5] discussed the contribution of virtual reality, particularly the Nintendo Wii, in exercise training and rehabilitation.

- Documentary (2024) [3] explored the potential and pitfalls of new quantum computers.

- Twilio (2025) [9] provides documentation on programmable messaging for developers.

References

[1] Al-Ansi, Abdullah M. et al. "Analyzing augmented reality (AR) and virtual reality (VR) recent development in education". In: *Social Sciences Humanities Open* 8.1 (2023).

[2] Boulton, Clint. “Can synthetic data help us scale AI’s data wall?” In: *The Next Platform* (2025). URL: https://www.nextplatform.com/2025/01/16/can-synthetic-data-help-us-scale-ais-data-wall/.

[3] Documentary, DW. *New quantum computers - Potential and pitfalls*. 2024. URL: https://www.youtube.com/watch?v=0HFzTYlhT2E.

[4] Gao, Minyu et al. “TETRIS: Scalable and efficient neural network acceleration with 3D memory”. In: *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*. 2017.

[5] Hiraga, Cynthia Y. et al. “Contribution of virtual reality (Nintendo Wii) for exercise training and rehabilitation”. In: *Locomotion and posture in older adults*. Cham Springer, 2024.

[6] Kuszmaul, William. “Randomized data structures: New perspectives and hidden surprises”. PhD thesis. Massachusetts Institute of Technology. Department of Electrical Engineering and Computer Science, 2023.

[7] Overleaf. *Bibliography Management with bibtex*. accessed 2025-02-11. 2025. URL: https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex.

[8] Pruksachatkun, Yada, Mcateer, Matthew, and Majumdar, Subho. *Practicing trustworthy machine learning: consistent, transparent, and fair AI pipelines*. O’Reilly Media, 2023.

[9] Twilio. *Programmable messaging*. accessed 2025-02-26. 2025. URL: https://www.twilio.com/docs/messaging.

[10] Wikipedia. *BibTeX*. accessed 2025-02-11. 2025. URL: https://en.wikipedia.org/wiki/BibTeX.