Git and LaTeX Worksheet

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?
   Answer: A version control system is a system that records changes to a file or set of files over time. Useful to allow you to track changes, collaborate, and revert to previous versions.
2. What is the difference between git and GitHub?
   Answer: Git is a version control system, while GitHub is an actual platform that hosts the repositories.
3. What is a repository?
   Answer: A repository is a directory where git tracks your file changes.
4. What is a commit?
   Answer: A commit is a snapshot of changes made to a git repo at a specific point in time.
5. What is the commit graph?
   Answer: Commit graph is a representation of the history of commits in a repository.
6. What is your preferred local git client (eg., command line, GitHub Desktop, GitKraken, etc.)?
   Answer: GitKraken, but I can also use the command line depending on the scale of the project.

2.2 Local Usage

1. What is the difference between adding a file to the staging area and committing a file?
   Answer: Adding a file to the staging area prepares the file to be committed, committing a file actually saves the changes to the repository.
2. What is a commit message, and why is it important for them to be meaningful?
   Answer: Commit message is a message that the committer writes to describe the changes. It is important for them to be meaningful so that collaborators (and the original committer) can understand the changes made.
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
   ```

   Answer:

   ```
   git init
   git add file.txt
   git commit -m "commit a"

   git commit -am "commit b"

   git commit -am "commit c"

   git commit -am "commit 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.
   Answer:

   ```
   git checkout BBBBBB
   git checkout -b recover-branch 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?

Answer:
Technically, if nothing is committed, the only thing you can really do is just try and fix your file. However, we can try a couple things that might work: checking stashed changes, checking reflog, or manually modifying a single file. Here are those three methods, in order:

1.

```
git stash list
git stash apply
```

2.

```
git reflog
#one of the below lines
git checkout HEAD@{1}
git reset --hard HEAD@{1}
```

3.

```
git checkout -- file.txt
```

2.3   Branching and Merging

1. What is a branch? Why are they useful?
   Answer: A branch is a pointer to a specific commit. They are useful because they allow you to work on different features without affecting the main branch. They also let multiple people easily work on seperate features of the same project.

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
```

Answer: Here the commands from git command line with basic messages:

```
git init
git add file.txt
git commit -m "Commit A"
```

```
git commit -am "Commit B"

git branch feature-branch
git checkout feature-branch

git commit -am "Commit E"

git commit -am "Commit F"

git checkout main
git commit -am "Commit C"

git commit -am "Commit D"
```

3. Why is a merge? Why are they useful?
   Answer: A merge is the process of combining changes from one branch to another. They're useful because they allow you to combine changes from different branches, i.e. two seperate people can work in two branches then combine their changes.

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---/
```

Answer: Assume we are starting from commit D.

```
git checkout feature-branch
git commit -am "Commit F"

git checkout main
git merge feature-branch -m "m main"
git commit -am "Commit G"
```

5. What is a merge conflict? When do they occur?
   Answer: A merge conflict occurs when Git can't automatically fix differences between two branches. The most common time this happens is when the same lines in a file are changed in both branches. This requires a manual fix.

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.
Answer:

```
git init
git add conflict.txt
git commit -m "Commit A"

git branch feature-branch
git checkout feature-branch

git commit -am "Commit B"
git checkout main
git commit -am "Commit C"
git merge feature-branch
```

2.4 Remotes

1. What is a remote?
   Answer: A remote is a repository that is not on your local machine.
2. What does pushing and pulling do?
   Answer: Pushing sends your changes to the remote repository, pulling gets the changes from the remote repository.
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?
   Answer: Just clone the repo from GitHub. Unfortunately, you'll lose the changes made between the last push and the death of your laptop.

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) [6] 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.
Answer:
`\textcite{}` integrates the citation into the text by printing the author's name as part of the sentence (and the year in parentheses). More useful when you want to emphasize the authors contribution. In contrast, `\cite{}` produces a citation that appears in parentheses. This form is useful when you want to support a statement without interrupting your flow. I often put these at the end of a sentence as a sort of supporting evidence for a claim I've said that's backed up by the source.

Here are examples from my sources:

- **Journal Article:** "Einstein (1905) [1] laid the foundations of special relativity."
- **Conference Article:** "The study by Morris (1970) [5] on storage techniques has influenced..."
- **PhD Thesis:** "As demonstrated in the work of Knuth (1968) [3], algorithm analysis is a key part of computer science."
- **Article in Popular Media:** "Recent discussions on digital media trends can be found in [4], where the impact..."
- **Book:** "Even the ideas presented in Turing (1950) [8] still influence computer science today."
- **Chapter of a Book:** "Information theory was revolutionized by [7], who introduced the concept of..."
- **YouTube Video:** "For an explanation of temperature misconceptions, see Veritasium (2012) [9]."
- **Technical Documentation:** "Please refer to [2] for up-to-date information on syntax and libraries."

References

[1] Einstein, Albert. "Zur Elektrodynamik bewegter Körper". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: `10.1002/andp.19053221004`.

[2] Foundation, Python Software. *Python 3 Documentation*. Accessed: 2021-09-01. 2021. URL: `https://docs.python.org/3/`.

[3] Knuth, Donald E. "The Art of Computer Programming, Volume 1: Fundamental Algorithms". PhD thesis. Stanford University, 1968.

[4] Monroe, Roberto G. "The Algorithm That Makes Preschoolers Obsessed With YouTube". In: *The Atlantic* (Nov. 2015). URL: `https://www.theatlantic.com/technology/archive/`

2015 / 11 / the - algorithm - that - makes - preschoolers - obsessed - with - youtube/414163/.

[5] Morris, Robert. "Scatter Storage Techniques". In: *Proceedings of the 1970 ACM SIGFIDET (now SIG-MOD) Workshop on Data Description, Access and Control*. SIGFIDET '70. 1970, pp. 6–13. DOI: `10 . 1145/1734663.1734671`.

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

[7] Shannon, Claude E. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal*. Vol. 27. 3. Bell Labs, July 1948, pp. 379–423. DOI: `10 . 1002 / j . 1538 - 7305 . 1948 . tb01338.x`.

[8] Turing, Alan M. *Computing Machinery and Intelligence*. Vol. LIX. 236. Mind, 1950, pp. 433–460. DOI: `10.1093/mind/LIX.236.433`.

[9] Veritasium. *Misconceptions About Temperature*. YouTube. Mar. 2012. URL: `https : / / www . youtube.com/watch?v=vqDbMEdLiCs`.

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