
General Instruction: To complete the homework set, you are required to do the followings. Your solutions must be typed in \LaTeX using the course homework template. The progression of your homework solution is to be “recorded” by making a git folder specifically for this homework set. The burden of proof is on you, and if your git commit history is sparse, then you may be liable for a penalty. A paper copy of the PDF output of your \LaTeX file is to be submitted to your instructor in class on the due date. *After* submitting the paper copy, but *before* the end of the due date, you will upload your work to your github by making a remote repository specifically for the homework, and post the link to the repository at the designated *Discussion* forum in Blackboard by making a thread just for you. The repository name in your github should be `550400.homeworkset.1` and the discussion forum thread should be named `YourFirstNameMiddleInitialLastName`, e.g., `BarackObama` and `WillardMRomney`. You have till the end of the due date to finalize your github repository. However, any commit made after the class time of the due date will be inadmissible. *Your attention to details in following this instruction will be critical, and if not followed exactly at the time of collection, the homework set may be graded at 90% of the full score.*

Problem 1 (10 pts): Assume that you are starting from “scratch” at the directory `~/`. Provide a sequence of git/bash commands that yields a git folder with a commit history such that:

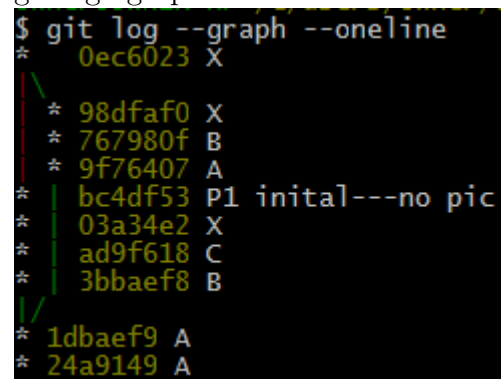
- the *master* branch has commits *A*, *B*, *C*, *X* and *D*,
- the *alt* branch has commits *A*, *B*, *X*,

Suppose that you are currently working on `master` branch. Draw its commit history graph (i.e., the graph portion of the output of `git log --graph --oneline`). Next, assume that you are on `alt` branch. Draw its commit history graph.

Problem 1 Solution:

```
Git command list: mkdir 550400.homeworkset.1.git
cd 550400.homeworkset.1.git
git init
vi main.txt
git add .
git commit -m "A"
git branch alt
git checkout alt
vi main.txt
git add .
git commit -m "A"
vi main.txt
git add .
git commit -m "B"
vi main.txt
git add .
git commit -m "X"
```

```
git checkout master
vi main.txt
git add .
git commit -m "B"
vi main.txt
git add .
git commit -m "C"
vi main.txt
git add .
git commit -m "X"
git merge alt
vi main.txt
git add .
git commit -m "D"
git log --graph --oneline
```



```
$ git log --graph --oneline
* 0ec6023 X
/
* 98dfaf0 X
* 767980f B
* 9f76407 A
|
* bc4df53 P1 initial---no pic
|
* 03a34e2 X
|
* ad9f618 C
|
* 3bbaef8 B
/
* 1dbaef9 A
* 24a9149 A
```

The screenshot shows a terminal window with the command `git log --graph --oneline` executed. The output displays a commit graph with branches and merges. The graph shows a sequence of commits: `0ec6023 X` (root), `98dfaf0 X`, `767980f B`, `9f76407 A`, `bc4df53 P1 initial---no pic`, `03a34e2 X`, `ad9f618 C`, `3bbaef8 B`, `1dbaef9 A`, and `24a9149 A`. The graph indicates that `0ec6023 X` is the parent of `98dfaf0 X`, `98dfaf0 X` is the parent of `767980f B`, `767980f B` is the parent of `9f76407 A`, `9f76407 A` is the parent of `bc4df53 P1 initial---no pic`, `bc4df53 P1 initial---no pic` is the parent of `03a34e2 X`, `03a34e2 X` is the parent of `ad9f618 C`, `ad9f618 C` is the parent of `3bbaef8 B`, `3bbaef8 B` is the parent of `1dbaef9 A`, and `1dbaef9 A` is the parent of `24a9149 A`. The graph also shows a merge from `3bbaef8 B` to `1dbaef9 A`.