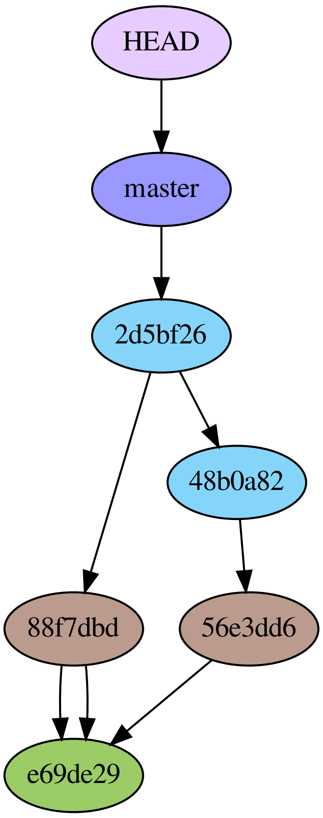
Department of Computer Science and Engineering
Indian Institute of Technology Bhilai
CS200 – Software Tools and Technologies Lab II
Scope: Git Object Model/Git Branching/Git Remote
Difficulty Level: Intermediate

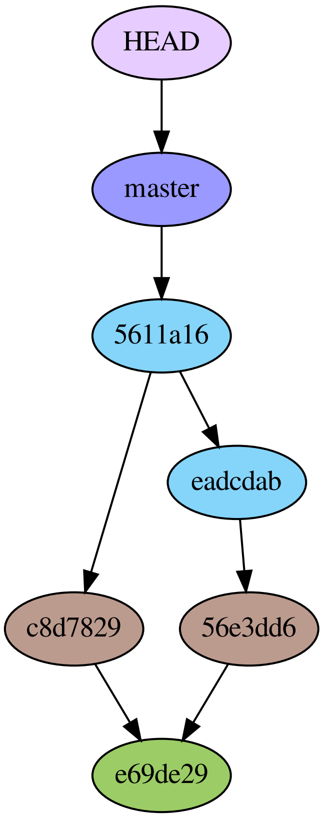
Assignment 2
February 23,2023

- Instructions
  - All answers will be in separate files in a single folder, named: `<group-id>_<group-name>`
  - Name files as `q<question-no>` without any extension. e.g., q2
  - Use LaTeX to show your answers that need `git` graphs
  - Make a `tarball` for the folder that contains your answers
  - Compress the `tarball` using `gzip` before uploading on Canvas

1. What is the difference between the following repositories? [Warm Up!]

   Try to recreate both. Now comment on the evolution of the second repository.





2. Develop a `git` flow with 10 successive commits. [Branch it On!]

   - Write an *iterative* shell script that uses ~ operator to create a branch at every ancestor of the last commit.
   - Write a *recursive* shell script that uses ^ operator to create a branch at every ancestor of the last commit.

3. Demonstrate a case where the branch merging in `git` leads to a conflict. [Merge Conflicts]

   Show how we can visualize the conflict. Resolve the conflict and show that the subsequent attempt to merge succeeds.

4. There is a way to `commit` a file without staging it. Find it. [Commit without staged!]

   Now explain how this works in practice. Can you demonstrate this with an example. If so write a script for the same.

5. Find out the usage of `git commit --amend` other than the one discussed in class. [Amend-Meant]

   Now show one example each of both the usages of `amend` with respective `git graphs`.

6. Write a shell script to auto-make a `git` repository with 50 commits. [Lost and Found]

   - Every commit should contain a new file.
   - Every commit will have a commit message "This is commit number `<Commit-count>`".
   - Now use the `awk` command to exact the list of commits from the `git log`
   - Now use `awk` again to randomly pick one of the commits
   - Can you merge the above two steps using a single invocation of `awk`?
   - Finally `checkout` a branch from the picked commit and share the git graph (that shows *only* the commits).

7. Write a shell script to auto-make a `git` repository with 50 commits. [`grep` it out!]

   You can *reuse* the script from the last question.

   - Modify the script to give a random time-gap of $1 \rightarrow 5$ seconds between every commit.
   - Now use the `grep` command to make a time-line of commits from the `git log`.

8. Recall the website shared in class: `http://ndpsoftware.com/git-cheatsheet.html` [The Cheatsheet]

   Pick one command from the cheatsheet that was *not shown in class* and demonstrate its usage through a `git` flow.

9. Create a private repository on `github` and make the first commit. [`git` remote]

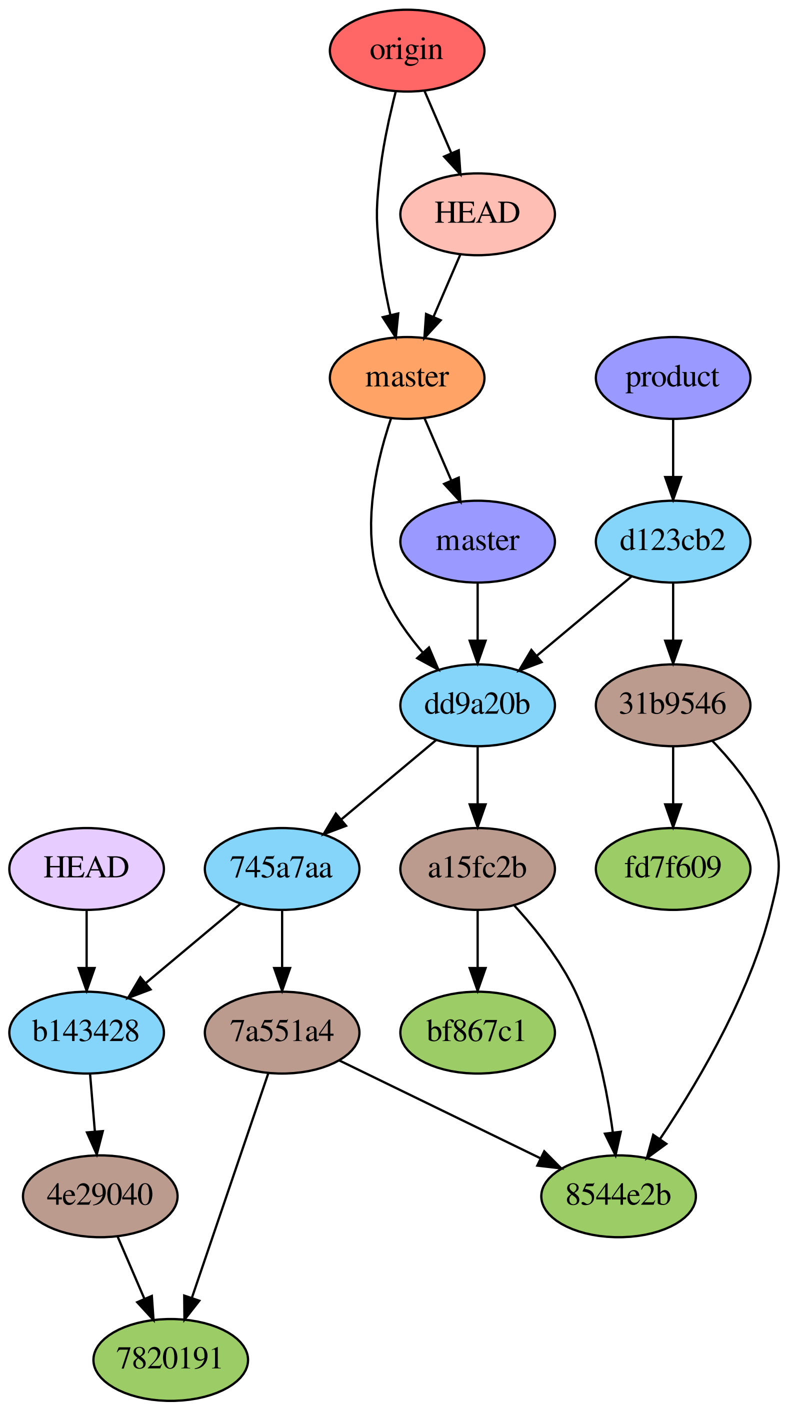
   Then do the following.

   - Add your group members as collaborators (who should accept the invitation)
   - All members should clone the repository on their systems.
   - All members should create a branch named `<Roll-No>` and make two commits on them.

   Find out how you can let the other members of your group know about the branch you created *locally.*

   After all branches are synced with remote share the `git graph` of each member.

10. Recreate the following `git graph`.


origin
HEAD
master
product
master
d123cb2
dd9a20b
31b9546
HEAD
745a7aa
a15fc2b
fd7f609
b143428
7a551a4
bf867c1
4e29040
8544e2b
7820191