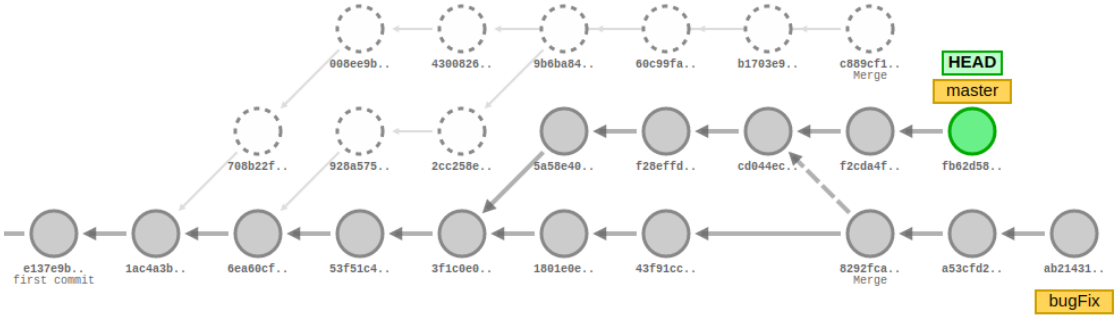
DEPARTMENT OF EECS
Indian Institute of Technology Bhilai
CS200 — SOFTWARE TOOLS AND TECHNOLOGIES Lab II
Scope: Git Branch
Difficulty Level: Advanced

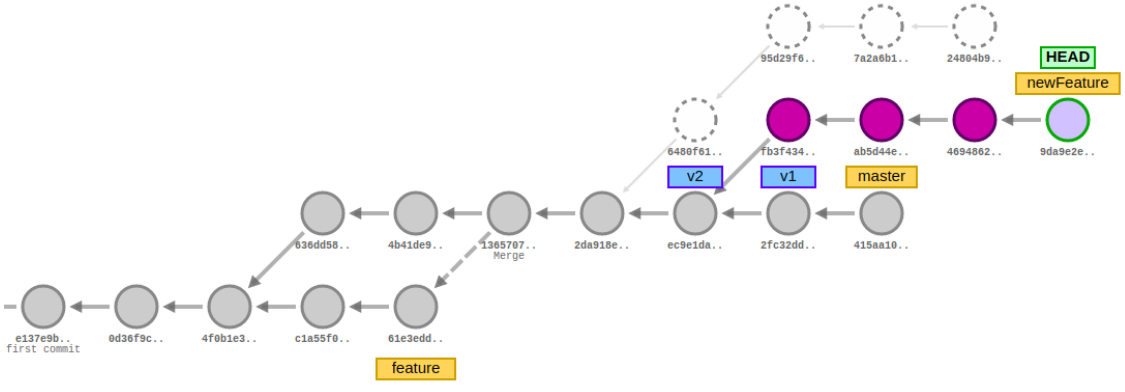
Assignment 3
October 17, 2020

- 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 Piazza

1. (a) List down the abstract `git` commands that lead to the following in `git` visualizer in `https://git-school.github.io/visualizing-git/#free`



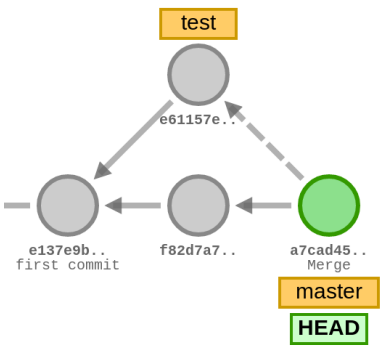
(b) List down the abstract `git` commands that lead to the following in `git` visualizer in `https://git-school.github.io/visualizing-git/#free`

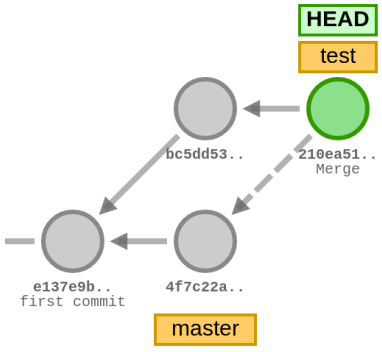


(c) The `merge commit` is a special `commit` object. Find out the contents that go into calculating the `shasum` of such a commit.

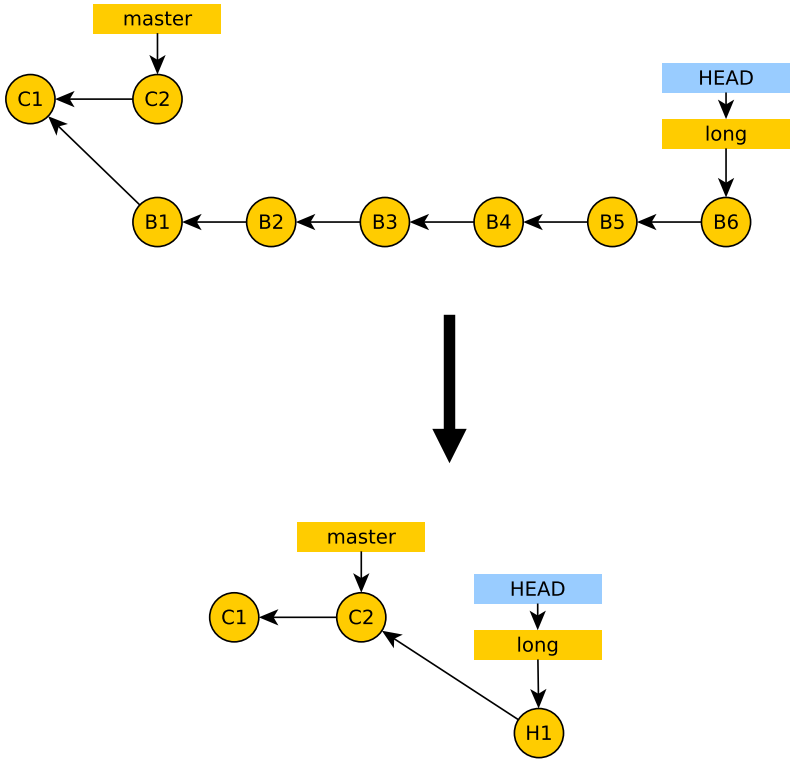
2. Recall that the `merge commit` object has multiple parents.

   What will be the output of the execution of the following `git` relative reference command: `git checkout HEAD^` on each of the `git` graphs below. Show the resultant `git` graphs. Now explain how `git` resolves this ambiguity. [Note: The graphs themselves give you a hint]





3. Find the `git` command that will lead to the following transformation for the `git` commit tree. Demonstrate the same with an example `git` flow.



4. Recall the Problem 4 of the Assignment 2. Please proceed until step (e) as it is, and then, try the following steps with the condition that *you cannot commit any unfinished code in any branch at any point of time.*

 Question 4 continues on the next page...

- (f) In the same "master" branch, start updating the code for *zigzag* traversal in such a way that it prints the reverse *zigzag* now (bottom to top).
- (g) Assume that you have an urgent requirement of building a new functionality (converting a given binary tree into a binary search tree) before finishing the last step. Create another branch "NewFunc" and write the code for implementing the new functionality. Complete it and commit the "NewFunc" branch.

(i) At this stage, please write the commands to go back to the "master" branch, retrieve and finish the unfinished traversal code, commit and then merge the "NewFunc" branch with "master". Showing only the `git` commands would be sufficient; no need to submit the traversal or tree conversion codes.

(ii) Suppose, you were simultaneously working on three different functionalities at the "master" branch when you had to shift to the "NewFunc" branch. And then, after returning back to the "master" branch, you want to finish two of them and remove the third one. Please write the required commands for doing that.
