Code and data management with Git

The git commit graph

1 General questions

1. A git branch is:

   (a) a file
   (b) always a problem
   (c) a pointer to a commit
   (d) a merge conflict
   (e) where leaves could grow
   (f) none of the above

2. How can you see the commit history in a tree alike manner?

   (a) git state
   (b) git repository
   (c) git commit
   (d) git status –tree
   (e) git show tree
   (f) git log –graph

3. What command could be used in joining two or more development histories (branches) together?

   (a) git join
   (b) git combine
   (c) git merge
   (d) git pull
   (e) git fork
   (f) git intersect

4. Which of the following commands displays the branches list?

   (a) git show-branch
   (b) git merge
   (c) git display
   (d) git branch
   (e) git list
   (f) git clone

5. Which of the following creates a new branch called brand_new and also checks it out?

   (a) git branch -c branch_new
   (b) git new branch branch_new
   (c) git status brand_new
   (d) git checkout -b brand_new
   (e) git commit brand_mew
   (f) none of the above

6. In order to merge a branch do we have to add and commit, or discard any changes first?

   (a) yes
   (b) no

7. If we want to merge brand_new into master, which branch should be checked out when we run the command git merge brand_new?

   (a) brand_new
   (b) any branch
   (c) master
   (d) none of the above

8. Which of the following commands will delete the branch brand_new?

   (a) it is impossible to delete a branch
   (b) git delete branch brand_new
   (c) git head -d brand_new
   (d) git branch -d brand_new

9. You are in a detached head state when:

   (a) there is no head
   (b) the head points to the master branch
   (c) the head points to a commit
   (d) the head points to a different branch then the master branch

2 Inspecting the commit graph

In the following you will work with the repository present in `∼/commit-graph-practical`. Please navigate to that directory and try to use the commands presented during the lecture (with their options accordingly) to perform the following tasks and answer the corresponding questions.

1. How many branches are there?

   (a) none

   (b) only the master branch

   (c) 4

   (d) 3

2. Are you in a detached head state?

   (a) yes

   (b) no

3. What branch is active (checked out)?

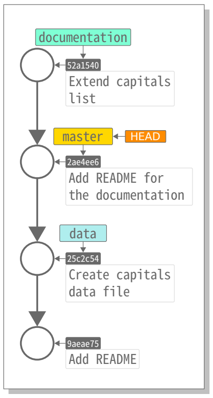
   (a) data

   (b) data-europe

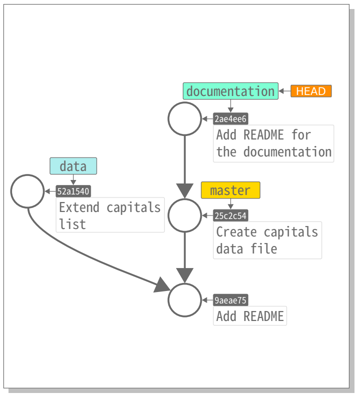
   (c) documentation

   (d) master

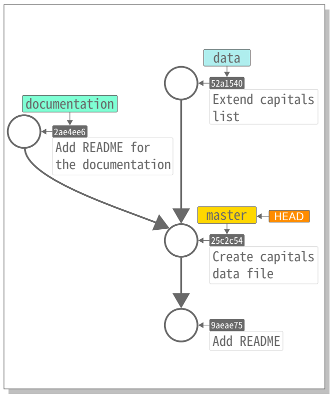
4. The commit graph looks like:



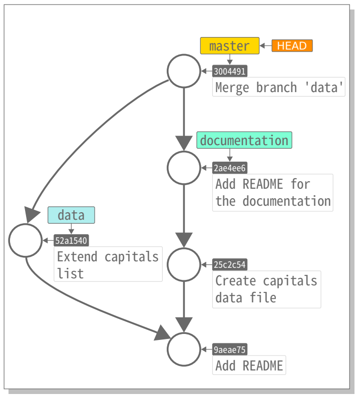
(a)



(b)



(c)



(d)

5. What merge type is performed if branch `documentation` is merged into `master`?

   (a) fast-forward

   (b) try-and-leave

   (c) three-way

   (d) advanced

6. Merge branch `documentation` into `master`.

7. What merge type is performed if branch `data` is merged into `master`?

   (a) fast-forward

   (b) three-way

   (c) try-and-leave

   (d) advanced

8. Merge branch `data` into `master`.

3 More commit graph manipulation

You have already performed some merging in the previous section, but the repository state was beforehand prepared for you. Next, you will make some file changes and manipulate the commit graph even further.

3.1 Fast-forward merging

1. Checkout branch `data` and create a new branch called `data-europe`.
2. Checkout branch `data-europe`.
3. Edit the `data/capitals.txt` file by adding at the end of it Rome as Italy's capital ("Italy:Rome"). Do not remove or make any changes to the information already present there.
4. Stage and commit the changes.
5. Checkout branch `data` and merge the `data-europe` branch into it.
6. Why was the previous merge a fast-forward one?
   (a) it was not
   (b) the branches did not diverge yet
   (c) different commits
   (d) none of the above

3.2 Three-way merging without conflicts

1. Make sure branch `data` is active (checkedout).
2. Correct the USA capital to Washington D.C. in the `data/capitals.txt` file.
3. Stage and commit the changes.
4. Checkout branch `data-europe`.
5. Add Paris as capital of France at the end of the `data/capitals.txt` file.
6. Stage and commit the changes.
7. Checkout branch `data` and merge into it the `data-europe` branch.

3.3 Resolving a merge conflict

1. Make sure branch `data` is active (checkedout).
2. Add Moscow as capital of Russia at the end of the `data/capitals.txt` file.
3. Stage and commit the changes.
4. Checkout branch `data-europe`.
5. Add Madrid as Spain's capital at the of the `data/capitals.txt` file.
6. Stage and commit the changes.
7. Checkout branch `data` and merge into it the `data-europe` branch.
8. Solve the merge conflict that was created during the previous step.

3.4 Branch deletion

1. Delete the `data-europe` branch.