[CS200]- Software Tool and Technologies Lab-II **15-10-2020**

Homework: 3

Team Name: Peaky Blinders Roll Numbers:11940150, 11940370, 11940380

Solution of Problem 3:

We are given with a git commit tree. Now the task is to find out all the git commands that will end up in the given git commit tree.

Explaination :- Feature used will be the squash feature of **interactive git rebasing.**
We will 1st switch to the branch named **long** and after that we have to squash the commits from B1 to B5 into B6.

Given below is the **Git Commands** used to get the output

```
mkdir hw3Test
cd hw3Test
git init
echo ”Hello” >> f1
git add f1
git commit -m ”f1 added”
git branch long
echo ”Hello” >> f1
git add f1
git commit -m ”f1 mod”
git checkout long
echo ”Hello” >> f2
git add f2
git commit -m ”f2 added”
echo ”Hello” >> f3
git add f3
git commit -m ”f3 added”
echo ”Hello” >> f4
git add f4
git commit -m ”f4 added”
echo ”Hello” >> f5
git add f5
git commit -m ”f5 added”
echo ”Hello” >> f6
git add f6
git commit -m ”f6 added”
echo ”Hello” >> f7
git add f7
git commit -m ”f7 added”
git graph -n hlc
git rebase -i master
git graph -n hlc
```

In rebase pick first commit and squash all others
In after rebase you may see a few extra commits but they don't have any branch pointing to them so they are lost commits and are not being used
Ignoring these commits you can see that the git graph conversion is as asked in the question.
The command used is git rebase -i master

Given below is the git graph before rebasing

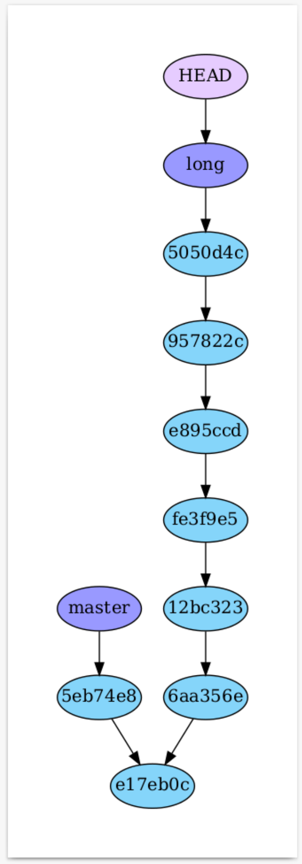

Figure 1.1: Git graph before rebasing

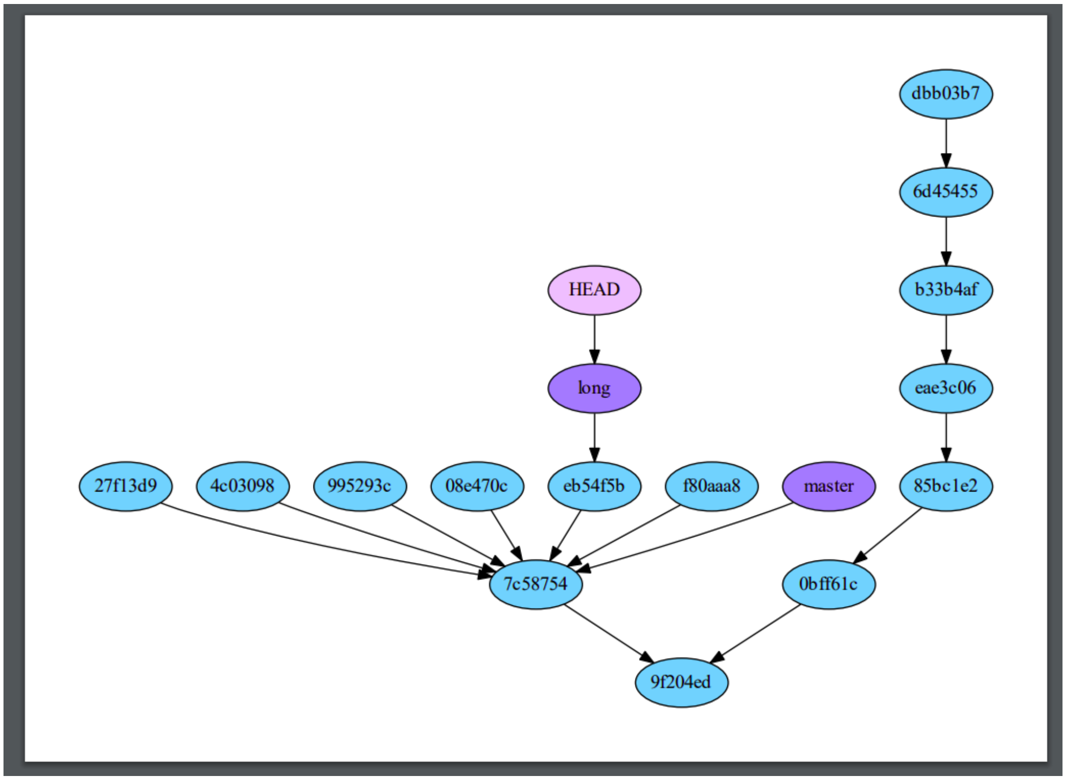

Figure 1.2: Git graph after rebasing