[CS200]-STNT-2 **Fall 2020-21**

Homework: 02

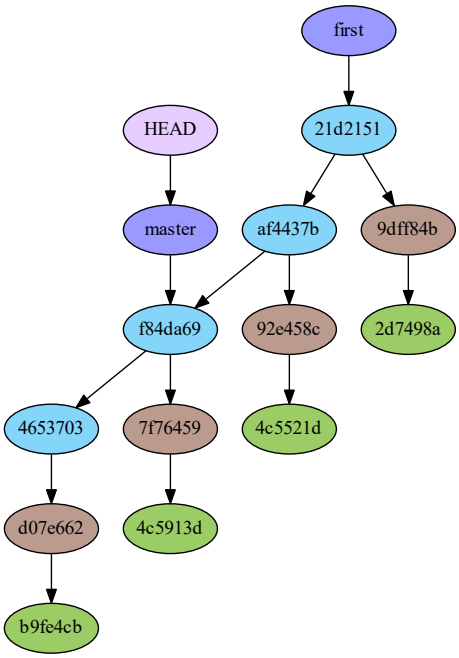
16. Technomakers: Anupam Kumar (11940160), Abdur Rahman Khan (11940020), Ruchit Prakash Saxena (11941040)

Solution of problem 3..
Git Shell script for merge is- $ git init
$ gedit test1.c
// Write void int main()
$ git add
$ git commit -m ”1st commit”
$ git branch first
$ gedit test1.c
// Write void merge()
$ git commit -m ”2nd commit”
$ gedit test1.c
// Write void mergesort()
$ git commit -m ”3rd commit”
$ git checkout first
$ gedit test1.c
// Write void print()
$ git commit -m ”4th commit”
$ git checkout master
$ git merge first
$git graph ...see figure 3.1

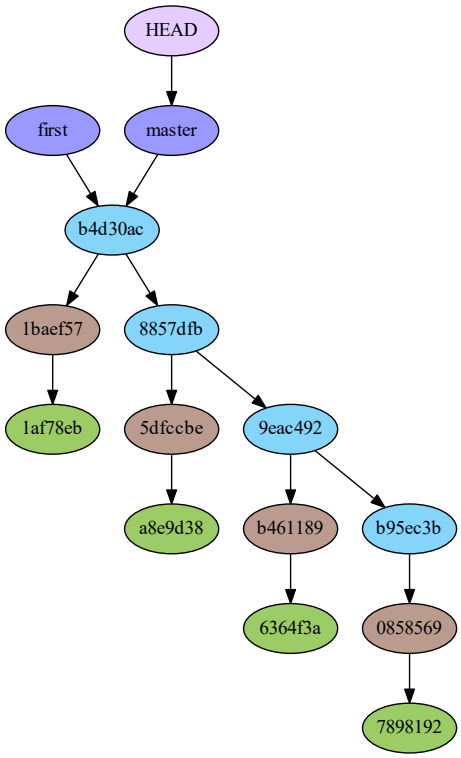
$ git init
$ gedit test1.c
// Write void int main()
$ git add
$ git commit -m ”1st commit”
$ git branch first
$ gedit test1.c
// Write void insertpivot()
$ git commit -m ”2nd commit”

```
$ gedit test1.c
// Write void insertionsort()
$ git commit -m ”3rd commit”
$ git checkout first
$ gedit test1.c
// Write void print()
$ git commit -m ”4th commit”
$ gedit test1.c
// Write edit insertionsort()
$ git rebase master
$ git graph                                    ...see figure 3.2
```

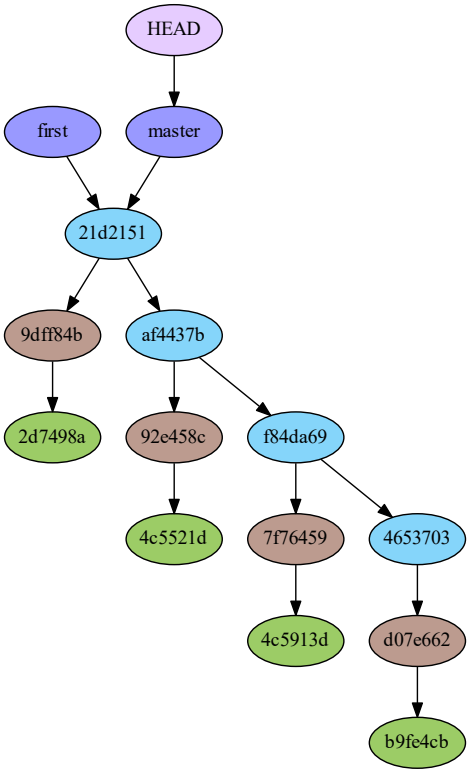
In first example (mergesort) as we want to see complete history of what happened, we should use merge feature, while in second example (insertionsort) we are only focussed on end file rather than history of it, so we use rebase feature of git. Merge preserves history whereas rebase rewrites it. Merging takes the contents of the feature branch and integrates it with the master branch. As a result, only the master branch is changed. The feature branch history remains same. As we rebase a feature branch onto master, we move the base of the feature branch to master branch's ending point. The history of commit how this file came to this version, get lost in rebase as it rewrites it.



(a) Figure 3.0: **Graph before merging or rebasing**



(b) Figure 3.1: **Graph after merging**



(c) Figure 3.2: **Graph after rebasing**