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

Homework: 01

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

Solution of problem 1. The commands for the given operations are as follows :

```
$ mkdir Question1
$ cd Question1
$ git init                                ..(a)
$ touch 11940160.txt                      ..(b)
$ echo "iitbhilai" >>11940160.txt
$ cp 11940160.txt 11940020.txt            ..(c)
$ cp 11940160.txt 11941040.txt
$ mkdir AnupamKumar                       ..(d)
$ mkdir AbdurRahmanKhan
$ mkdir RuchitPrakashSaxena
$ mv 11940160.txt AnupamKumar
$ mv 11940020.txt AbdurRahamKhan
$ mv 11941040.txt RuchitPrakashSaxena
$ git add .                               ..(e)
$ git commit -m "commited all the files at once "
$ git graph                               ..(f) [See figure 1.1]
```

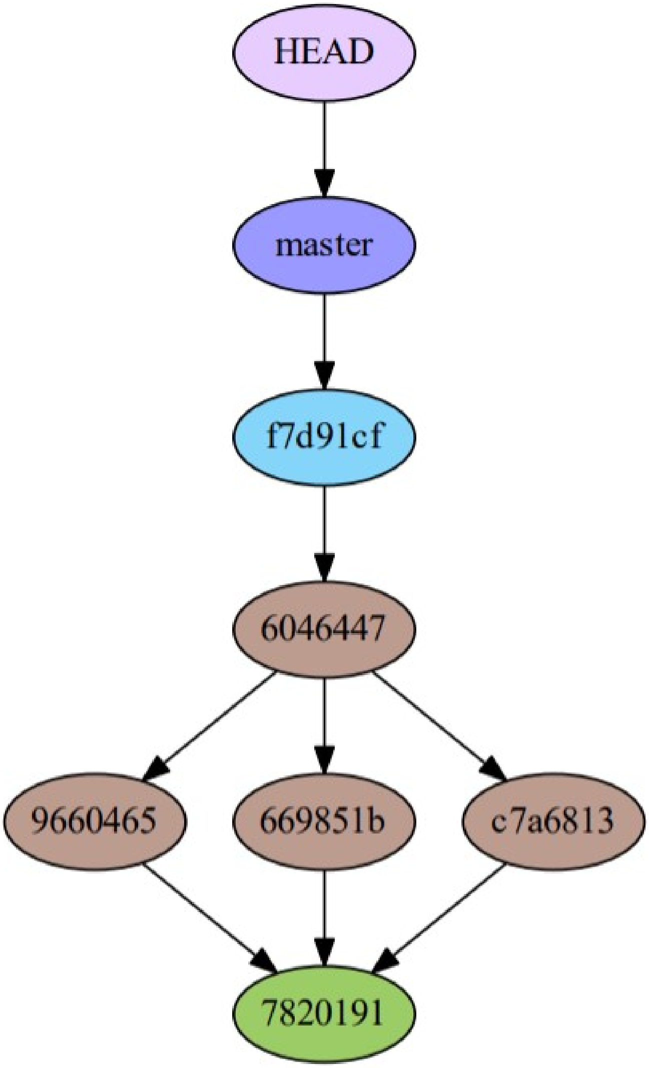
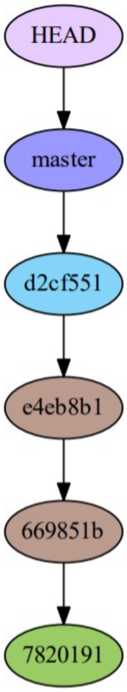


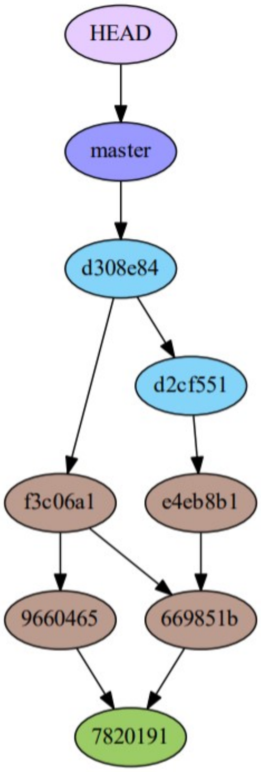
Figure 1.1: Question 1 - Git Graph

Solution of problem 2. The commands for the given operations are as follows :

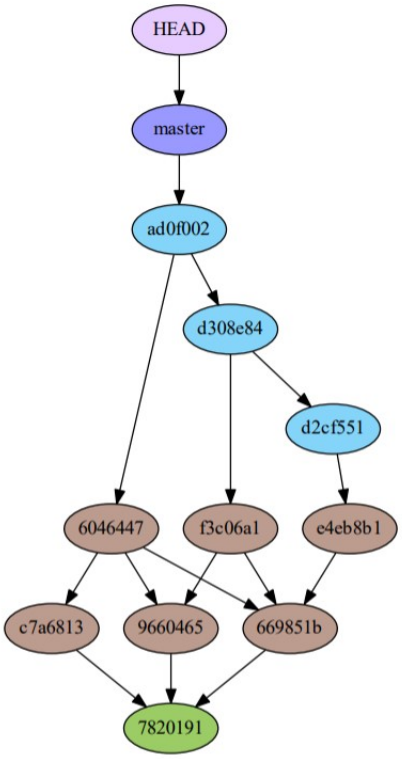
```
$ mkdir Question2
$ cd Question2
$ git init
$ touch 11940160.txt
$ echo ”iitbhilai” >>11940160.txt
$ cp 11940160.txt 11940020.txt
$ cp 11940160.txt 11941040.txt
$ mkdir AnupamKumar
$ mkdir AbdurRahmanKhan
$ mkdir RuchitPrakashSaxena
$ mv 11940160.txt AnupamKumar
$ mv 11940020.txt AbdurRahamKhan
$ mv 11941040.txt RuchitPrakashSaxena
$ git add AnupamKumar
$ git commit -m ”commited 1st member ”
$ git graph                                   .. [See figure 2.1]
$ git add AbdurRahmanKhan
$ git commit -m ”commited 2nd member ”
$ git graph                                   .. [See figure 2.2]
$ git add RuchitPrakashSaxena
$ git commit -m ”commited 3rd member ”
$ git graph                                   .. [See figure 2.3]
```



(a) Figure 2.1: **Graph 1**



(b) Figure 2.2 : **Graph 2**



(c) Figure 2.3 : **Graph 3**

Solution of problem 3. The required shell script is:

```
$ find .git/objects
        .git/objects
        .git/objects/info
        .git/objects/pack

$ find .git/objects -type f        : to show different types of objects in .git/objects(object database)
$ git cat-file -t < object >       : shows us the type of the object represented by a particular hash
$ git cat-file -p < object >       : shows the contents of the file associated with the hash value of
                                     object
```

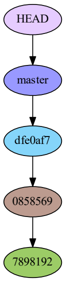
we can also use ⇒

```
$ git cat-file −−batch            : to print object information and contents for each object provided
                                     on stdin.
```

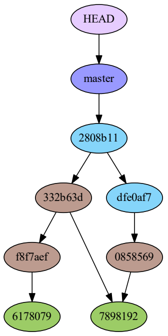
Solution of problem 4. .

The sequence of git commands that generates the given git graph are as follows:

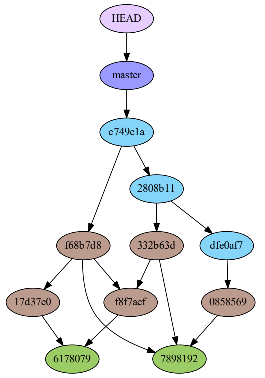
```
$ git init
$ touch a.txt
$ echo ”a”>a.txt
$ git add .
$ git commit -m”1st commit”
$ git graph                                   .. [See figure 2.1]
$ mkdir b
$ cd b
$ touch b.txt
$ echo ”b”>b.txt
$ git add .
$ git commit -m ”2nd commit”
$ cd ..
$ git graph                                   .. [See figure 2.2]
$ mkdir c
$ touch c.txt
$ git add .
$ cd ..
$ git commit -m ”3rd commit”
$ git graph                                        .. [See figure 2.3]
$ ls
$ git rm -r b
$ git commit -m ”removed a directory and 4th commit”
$ git graph                                        .. [See figure 2.4]
$ ls
$ git rm -r c
$ mkdir d
$ cd d
$ touch d.txt
$ touch e.txt
$ echo ”a”>d.txt
$ echo ”b”>e.txt
$ git add .
$ cd ..
$ git commit -m ”removed a directory and 5th commit”
$ git graph                                        .. [See figure 2.5]
$ ls
$ git rm a.txt
$ git commit -m ”removed a content and 6th commit”
$ git graph                                        .. [See figure 2.6]
$ ls
$ git rm -r d
$ mkdir f
$ cd f
$ touch f.txt
$ touch g.txt
$ echo ”a”>f.txt
$ git add .
$ cd ..
$ git commit -m ”removed a directory and 7th commit”
$ git graph                                        .. [See figure 2.7]
```



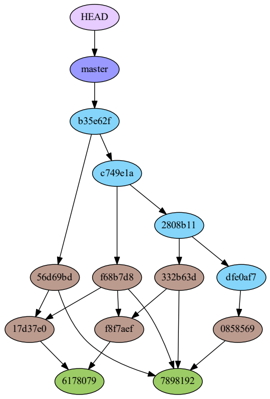
(a) Figure 2.1: **Graph 1**



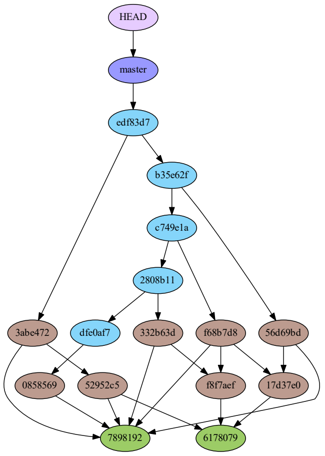
(b) Figure 2.2 :**Graph 2**



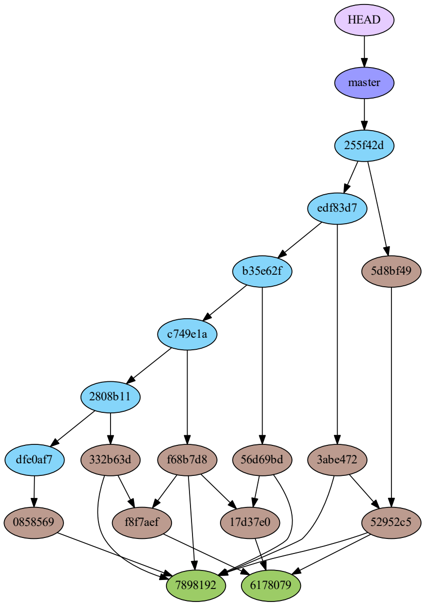
(c) Figure 2.3 : **Graph 3**



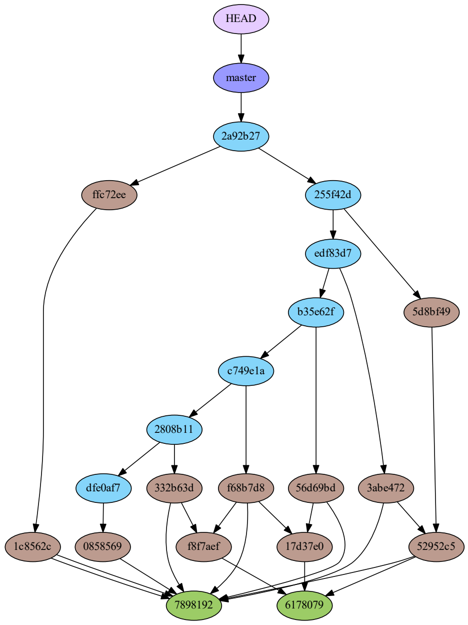
(d) Figure 2.4: **Graph 4**



(e) Figure 2.5: **Graph 5**



(f) Figure 2.6: **Graph 6**



(g) Figure 2.7: **Graph 7**