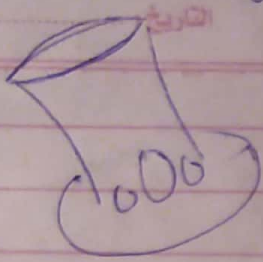


git diff only on [3] - a (src b.p) ~~early~~



Yehia Carak

git diff on [1] on [2]

sty ab +

++

3  
2  
1

1] Push.sh <sup>Python</sup> → Script & library in python to get ".git.log"

2] Call branch function (view) to get last global Commit hash from array of hashes →

Master [Master.length - 1] hash

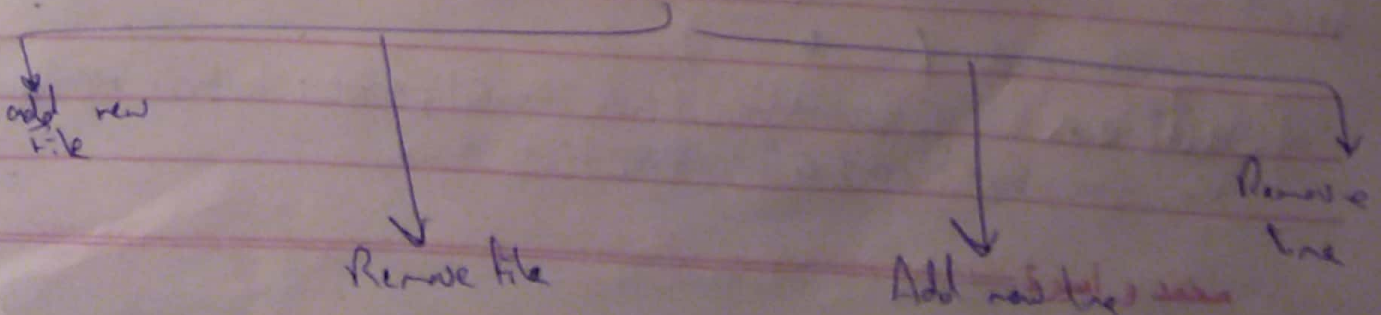
3] If the two hashes are equal ⇒ 5hs

4] If not equal ⇒ while (random not equal)

↳ Hunkeren el .git diff but el global be aser wished  
Tel array

↳ low mesh zgo benasod ne execute command  
lehad manasab we ben push fe array el  
Output ely ~~byetta3~~ byetta3 we fe aser  
Kol output brekkeh "end commit" zolichon  
Ne split beha

↳ Horemsky zala el array we hayeban Zandera  
4 Cases





# Solidity

Contract createRepo {

array repos[]  
map

→ function createRepo (constructor el new repo  
parameters (repoName, description), array of collaborators address  
})

Contract Repo {

name  
description  
owner  
collaborators[]  
date  
branches[]  
map []  
collaborator Map (address → bool)

→ Constructor (name, description, collaborators) {  
owner = msg.sender  
dateCreated = now  
parameters =

create newBranch, master = [map w array fortyeen]  
map ["master"] = masterBranch  
branches.push (master);

function addCollaborator (address collaborator) onlyOwner  
collab.push (collaborator)  
emit event CollabAdded (address coll, address, repoName)  
collaborator Map (collaborator) = true;

التاريخ:

الموضوع:

```
function removeCollab ( address, Coll ) only Owner  
  collab.Map( Coll ) = false  
  emit Collaborator Removed ( address, Repo Address, Name )
```

```
function makeBranch ( branchName ) only permitted {  
  branchName
```

```
  Branch newBranch = new Branch( , branches[0].map,  
    branches[0].array );
```

```
  branches.push( newBranch );
```

```
  map.push( newBranch );
```

```
  emit branchCreated ( repoName, branchName );
```



Branch

Contract Branch {

branchName,

CommitMap [CommitHash → Contract]

CommitArray [ ]

Constructor (branchName, CommitMap, CommitArray)

~~function makeCommit~~

function parseCommit(<sup>hash</sup>date, branch, msg, change) only permitted

Commit newCommit = new Commit(msg.sender,  
rets et fac parameters);

~~CommitMap~~ CommitMap.push(newCommit);  
CommitArray.push(newCommit);

emit CreatedCommit(msg.sender, msg, hash);

<sup>3</sup>  
~~function~~ function makeRoot (  
if (CommitArray.length - 1 == 0)

→ parseCommit( ) محمد وأميرة

\* Contract Commit {



function initCommit ( , EntProject)

change EntProject; }

getPartCommHash ( ) <sup>view</sup>

return commit [ commit.Length - 1 ] Hash; }