Select the right page in Gradescope or we will not grade the question!

Homework 5
Git II

Due: Wednesday, February 14th, 11:59PM (Hard Deadline)

Submission Instructions

For this assignment, all of your submissions will be to GitLab.

Optional Reading

Git is a purely functional data structure, by Philip Nilsson, from Jayway.

http://www.jayway.com/2013/03/03/git-is-a-purely-functional-data-structure/

I highly encourage reading this when you have some time to read the article carefully and think deeply about the material. This article presents an excellent way of thinking about git and how it operates.

1 Evaluating git usage

Earlier this semester, we asked you to use git with at least one project. Now you will set up that project to be shared with the course staff. Visit https://gitlab.eecs.umich.edu and create a new project named **exactly** `c4cs-w18-wk5`. Be sure to create this project as **Private** (not Interal or Public).

Add this new repository as a remote (`git remote add ...`) to your existing project.

Push the project to this new remote.

In GitLab, grant **Reporter** permission to `tarunsk`, `samkhan`, `amrith`, `cyanliu`, `sltries` and `mmdarden`. (Choose "Members" from the drop-down list from the settings gear in the top right to manage this permission).

We will run a test script (you can see the grading script here), checking our access to everyone's repository on Thursday.

Some key factors we are looking for:

1. Commit length. Not too long for the title of the commit message, and not too short. Also including more description in the body of the commit message is looked for.
2. Number of commits. Only having one or two commits in a repository doesn't mean you really used git effectively. In Homework 2 we asked you to begin using git for some sort of a project. This project should have at least five commits in it for it to be considered for full credit.

Again, for more insight, check out the script used to grade this homework assignment.

2 Handling merge conflicts

2.1 Content Conflict

Clone https://gitlab.eecs.umich.edu/c4cs/c4cs-git-conflict1.git

This repository has a master branch and a merge_me branch that have diverged. Merge the merge_me branch into master, resolving the conflict.

When you are done, running bash test.sh should print "Success" and running python main.py should print something reasonable (if factually inaccurate now, hooray!).

Create a new repository in your GitLab named **exactly** c4cs-git-conflict1. Be sure to create this project as **Private** (not Internal or Public).

Push your changes to your new repository. (*Note: This will be a different "remote"*)

In GitLab, grant **Reporter** permission to tarunsk, samkhan, amrith, cyanliu, sltries and mmdarden. (Choose "Members" from the drop-down list from the settings gear in the top right to manage this permission).

We will run a test script, checking our access to everyone's repository on Thursday night.

2.2 File Path Conflict

Repeat the same steps for https://gitlab.eecs.umich.edu/c4cs/c4cs-git-conflict2.git

Be sure to read over the commit history so that you are sure that the result of your merge has the right data!