AMS 550.400 HW SET 1 Yixuan DA

Last Compiled on October 10, 2012

General Instruction: To complete the homework set, you are required to do the followings. Your solutions must be typed in LaTeX using the course homework template. The progression of your homework solution is to be "recorded" by making a git folder specifically for this homework set. The burden of proof is on you, and if your git commit history is sparse, then you may be liable for a penalty. A paper copy of the PDF output of your LaTeX file is to be submitted to your instructor in class on the due date. *After* submitting the paper copy, but *before* the end of the due date, you will upload your work to your github by making a remote repository specifically for the homework, and post the link to the repository at the designated *Discussion* forum in Blackboard by making a thread just for you. The repository name in your github should be `550400.homeworkset.1` and the discussion forum thread should be named `YourFirstNameMiddleInitialLastName`, e.g., `BaracHObama` and `WillardMRommey`. You have till the end of the due date to finalize your github repository. However, any commit made after the class time of the due date will be inadmissible. *Your attention to details in following this instruction will be critical, and if not followed exactly at the time of collection, the homework set may be graded at* 90% *of the full score.*

Problem 1 (10 pts): Assume that you are starting from "scratch" at the directory `~/`. Provide a sequence of git/bash commands that yields a git folder with a commit history such that:

- the *master* branch has commits A, B, C, X and D,
- the *alt* branch has commits A, B, X,

Suppose that you are currently working on `master` branch. Draw its commit history graph (i.e., the graph portion of the output of `git log --graph --oneline`). Next, assume that you are on `alt` branch. Draw its commit history graph.

Answer

Command is listed as follows:

```
cd 550400
mkdir hw1
cd hw1
cd hw1
ls
git init
vi main.txt
git add .
git commit -m "A is done"
vi main.txt
git add .
git commit -m "B is done"
git branch
git branch alt
git checkout alt
vi main.txt
git add .
git commit -m "X is done"
git checkout master
vi main.txt
git add .
git commit -m "C is done"
git merge master alt
vi main.txt
git add .
git commit -m "clean up"
git merge master alt
git add .
git commit -m "Merge is done"
vi main.txt
git add .
git commit -m "D is done"
git checkout alt
cat main.txt
git push http://github.com/dayixuan/HW1 master
git push http://github.com/dayixuan/HW1 alt
git log –graph –oneline
history
```

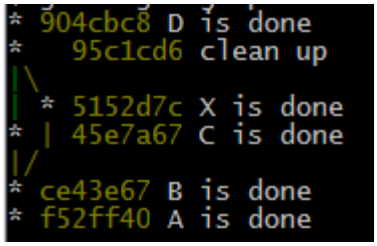

Figure 1: The history graph Master

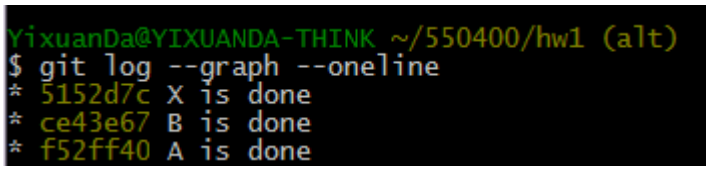

Figure 2: The history graph:Alt

Problem 2 (10 pts): Assume that you are starting from "scratch" at the directory `~/`. Provide a sequence of git/bash commands that yields a git folder and

- configure your git with your name and your email address,
- set up an alias for each of the git remotes listed below:

  ```
  git://github.com/nhlee/550400.stanza1.git
  git://github.com/nhlee/550400.stanza2.git
  git://github.com/nhlee/550400.stanza3.git
  ```

  Assume that each remote contains exactly single commit with a txt file for a single (different) stanza,
- pull to combine three stanzas of a poem,
- after the first pull, add the title of the poem,
- after the second and third pull, resolve the merge conflict,
- after resolving the third pull merge conflict, push the result to your (newly created) remote repository.

Answer

Command is as follows:

cd ..
mkdir hw2
cd hw2
git init
git config –global user.name ”dayixuan”
git config –global user.email ”dayixuan2010@gmail.com”
vi main.txt
git add .
git commit -m ”Alias is done”
git pull https://github.com/nhlee/500400.stanza1
vi main.txt
git add .
git commit -m ”P1 and title is done”
git pull https://github.com/nhlee/500400.stanza2
vi main.txt
git add .
git commit -m ”s2”
git pull https://github.com/nhlee/500400.stanza3
vi main.txt
git add .
git commit -m ”s3”
git push https://github.com/dayixuan/Homework1Problem2 master

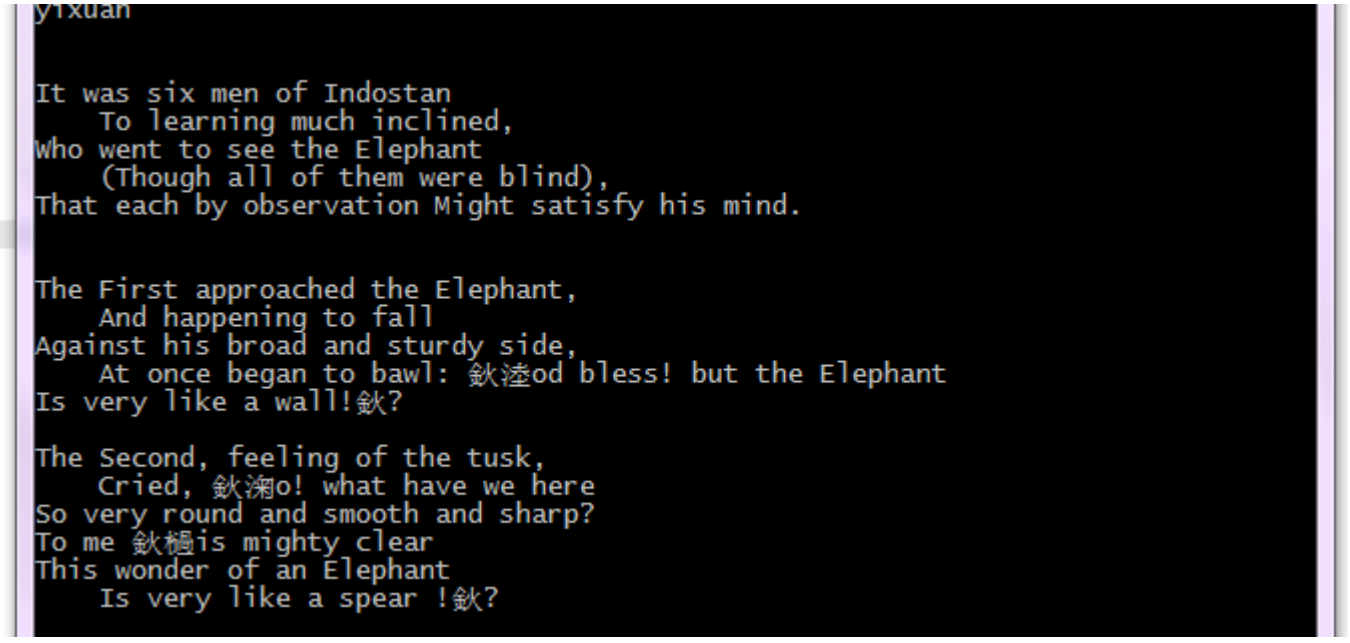

Figure 3: Poem

Problem 3 (40 pts): Consider a team of four students, say, A, B, C and D, who just started working on writing a `latex/beamer` file, say `main.tex`, for a class presentation of their work statement. Assume that they do not wish to coordinate their schedules for a concurrent group meeting (both virtually and physically). Assume that:

- A is in charge of *Introduction*,
- B is of *Problem Statement*,
- C is of *Timeline*,
- D is of *Deliverable* part of the presentation.

In other words, their contributions to `main.tex` do not overlap. Then,

- first, devise a work flow strategy for the team so that they can collaborate asynchronously using `git`,
- next, devise yet another `git` strategy different from your earlier proposal.

Finally,

- discuss the strength and weakness of each of your proposed strategies in terms of merge conflicts resolution,
- make the final recommendation.

In order to answer this question, *build* a mathematical model, *following* the guideline from IMM. Use Section 1.4 and Section 1.5 of IMM as *role models*. For example, you are to identify which variables are exogenous and which are endogenous. More specifically, among other things, in your model, is the preamble part of `main.tex` an endogenous or exogenous variable? Note also that in addition to this issue, there are other issues that you are to consider. So, *be sure to consult IMM*.

Answer

Strategy One:
Construct a git folder at the branch Master to start and after that four branches will follow, which represent four students: A for Introduction, B for Problem Statement, C for Timeline and D for Deliverable. When everyone finishes their own part, they will pull from the master, merge and then push to the master. This will prevent leaving out the others' work.

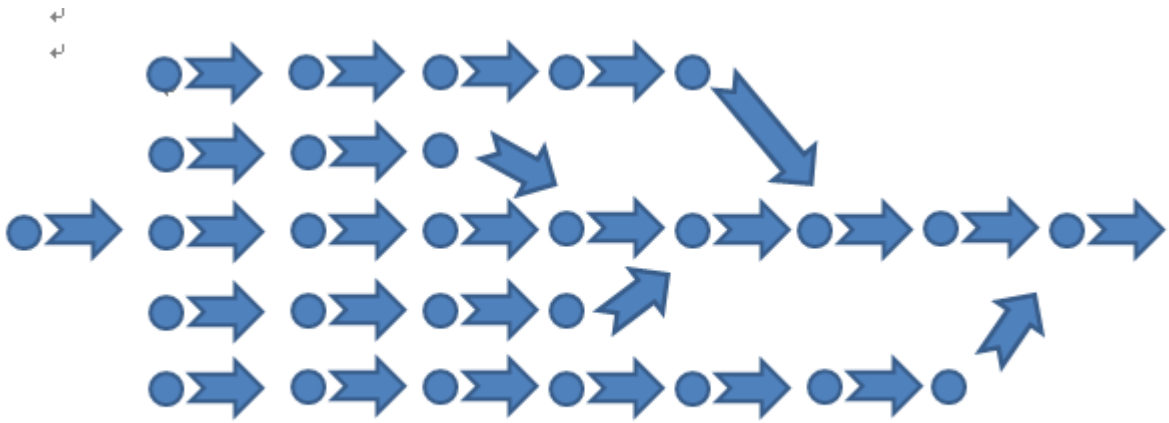

Figure 4: Strategy One

Strategy Two:
Everyone focuses on their own's work until they are done. After this, one of them may will construct an initial git folder on master and then do "pull and merge" for each of them.
Step1: Start with A Introduction, pull, add and commit.
Step2: Pull B's work Problem Statement, solve the merging conicts manually and add and commit.
Step3: Pull C's work Timeline, solve the merging conicts manually and add and commit.
Step4: Pull D's work Deliverable, solve the merging conicts manually and add and commit.

We only care about the final result: Introduction, Problem Statement, Timeline and deliverable. We do not focus on the other parts of main.txt which might be a short beginning or a conclusion. So those are considered exogenous variables in this context. Therefore we ignore the other parts of main.txt

For Strategy One, everyone focuses on their own branch and they don't interrupt each other. When they are ready to submit their parts, they will need to pull from the master and do a merging with other person's work. Then make a push to the master to include their work.In this way nobody's work will be left behind and each person will do a "pull, merge and push" to reach a final product.

For Strategy Two, each person is doing their part independently. When they are done, a random person will take the responsibility to pull everyone's work and merge manually.Finally that person will produce the last version of their group work.

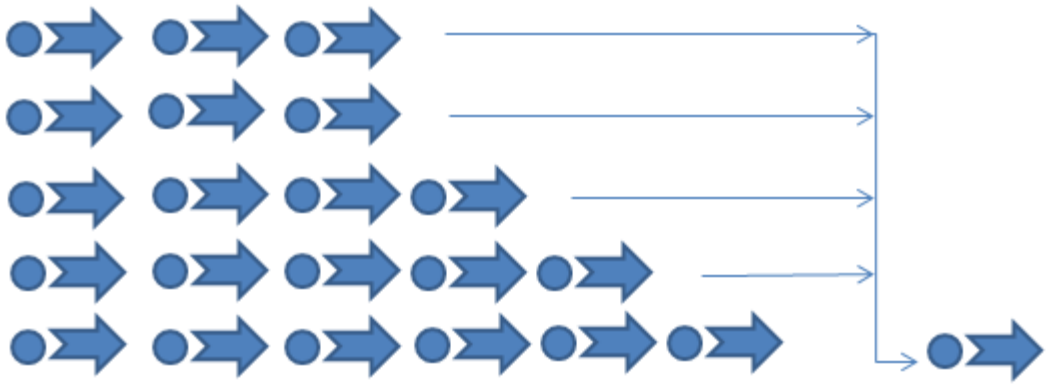

Figure 5: Strategy Two

Strategy Two is better because not all of them have to bother to do a series commands "pull, merge and push".There is no need to let everybody do repeatitive work.If someone wants to revise his work, he not only needs to revise but also have to do merge after revising, which will be time consuming and easy to make a mistake. Also, it will be quite easy for just one person to do pull, merge and push. Before submitting the final work,every body can revise their own. After this, the person who is resposible to summarize up the presentation will just command and produce.

To test the model we can simply assign short words for each part and then do a merge according to the above two strategies to assess which one is better in terms of merging issues.

Problem 4 (aka. Fair Play, 40 pts): Answer the following question:

Is the tennis game fair?

Note that unlike Problem 3, this question is vaguely stated. This is intensional, whence to begin, you will first need to clarify what exactly your question is. You may use the class discussion on this particular problem, but you *may not* directly refer to our discussion. Instead, formulate the model carefully but concisely in your own words.

Answer

Formulate the Problem
Our overall objective is to judge whether the game is fair or not and in order to do so, we need to clarify the rules of this tennis game and in what way we assess the fairness. In tennis game, the player who delivers the ball is called the server and the other player is the receiver. The server always goes first. If the server wins at first, he earns 15 points and then he will serve the ball again.If he wins the next, he will score 30, and scores 40 and 45 if wins next.If he wins 45, then he wins this round. If the the two player both score 40, one must win by two points in order to beat the other.

If Player A has better skills than Player B,then we can say that we believe that it is more likely for A to win the game. Also, the result depends on how the players psychological situation. The rule says that after each game, the server and the receiver will switch their role in next round.However,it is easy for most of the people to believe that the server has an advantage because he stands in a positive position and have more choices on the directions of the ball goes to, hence believing the game is not fair at all.

To check the fairness of the game, we basically need to examine the probability of winning the game for both the players are the same. So if the server wins the game with probability 0.5, therefore the receiver has the probability 0.5 too.Suppose an extreme case where the probability to win the game for the server equals to 1, which means the server will definately win this game. So the tennis game is not fair at all.

Outline the Model
Assumption:The two players are identical when competing with each other.
To do the statistical experiements, we randomly picked up M games in history between this two players. Count the numbers that the server wins: to do this, we write down 1 if a server wins the game. Then the probability to win a game for the server equals to the number of wins divided by the total number of games M.

We know that if increasing the sample size M, then we can narrow down our interval to an optimal level. Therefore, if the the estimate of the probability to win a game for the serve is very close to 0.5 we could conclude that the rule of the game is fair. If the estimate is far away from 0.5 we could say that each game favors the server. Then the server has a higher likelihood of winning the game than the receiver thus the rule of the game is unfair.

In Reality?
Theoretical we can test like this without considering other factors.But in the real world there are much more to be considered. The result will also be influenced by skills, players' familiarity with the court, players' nervousness and so on.

Test the Model
Collecting more data and test more paired experiments' result will help us know the accuracy of the above model.

Final Remarks about Problem 3 & Problem 4: They are open-ended problems. However, your scores will be determined by how well do you follow the exposition style outlined by IMM and WMA. For both problems, your write-up should be

- self-contained,
- covering all four parts of Section 1.3 of IMM,
- paying a particular attention to any causal relation that you might be investigating, following Chapter 3 of WMA,
- answering questions that are explicitly asked in the problem statements.

For Problem 3, focus mostly on Step 2 and Step 3 of Section 1.3 of IMM. For Problem 4, focus mostly on Step 1 and Step 2. For each problem, minimum 1 pages and maximum 2 pages.