

---

**General Instruction:** To complete the homework set, you are required to do the followings. Your solutions must be typed in  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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 `WillardMRomney`. 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.

The following git commands were used for this exercise:

```
mkdir Hw1.git
cd Hw1.git
git init .
vi main.txt (type A)
git add .
git commit -m "A"
vi main.txt (insert line after A and type "B")
git add .
git commit -m "B"
git branch alt
git checkout alt
vi main.txt ( add line after B and type "X")
git add .
git commit -m "X"
git checkout master
vi main.txt (add line after B and type "C")
git add .
```

```

git commit -m "C"
git merge alt
vi main.txt (resolve conflict)
git add .
git -m "merged"
vi main.txt (insert line after C and type "D" )
git add .
git commit -m "D"

```

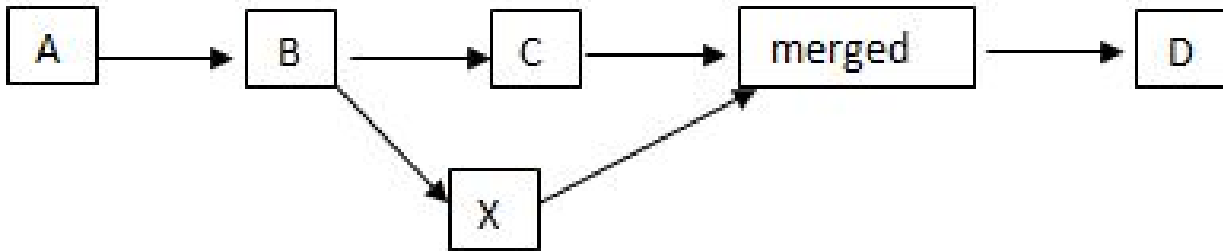


Figure 1: Master and branch

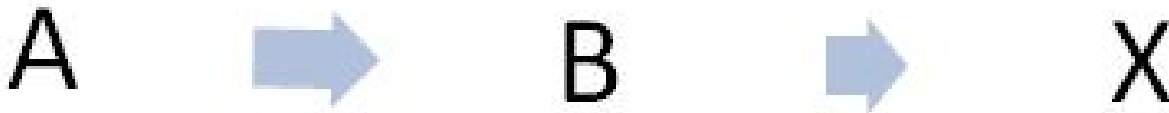


Figure 2: Alt Branch

**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.

The following sequence of git commands was used in this exercise:

```
mkdir Hw1Prob2.git
cd Hw1Prob2.git
git init .
git remote add stanza1 git://github.com/nhlee/550400.stanza1.git
git pull stanza1 master
vi main.txt (make sure it worked)
git add .
git commit -m "stanza1"
git remote add stanza2 git://github.com/nhlee/550400.stanza2.git
git pull stanza2 master
vi main.txt (resolve conflict)
git add .
git commit -m "stanza2"
git remote add stanza3 git://github.com/nhlee/550400.stanza3.git
git pull stanza3 master
vi main.txt (resolve conflict)
git add .
git commit -m "stanza3"
```

**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*.

**Proposal 1:** In this mechanism each student pushes their files onto their github.com account. Student B pulls from A and Student C pulls from B. Finally, Student D pulls from Student C. This allows for the files to merge without any additional copies being made. In addition, if the specific order is kept then the report will be in order and no further rearrangements will be required.

There is much strength to this mechanism. First, when the files are pulled by each student they are in order. Second, the probability of getting extra copies is at a minimum. There are some weaknesses, however. First, each student must be finished with their segment of the report before the next student can pull the file. The fact that the transfer of files must be in that specific order also makes the process less flexible. For example, A must be done before B, B must be done before C and C must be done before D. Another problem occurs with editing. For example, say that A wanted to change something in their section of the report. According to the mechanism, all the students would have to redo the entire process. This takes time and effort.

To test this we can perform a short simulation. Since I am only one person, it is tedious to establish four new accounts and act as each student. Instead, what I will do is I will use the poem exercise we did in class and act as each student. The first step is to assign a stanza to each student. For example, A will be stanza 1, B will be stanza 2, C will be stanza 3 and D will stanza 4. As B I

pull stanza 2 first. I then pull stanza 1 (A) and resolve the conflict. Next, I act as student C so I pull stanza 3. As student C I pull the file from B, which basically contains the file from A and B. As student C, I resolve the conflict. The same applies when I act as student D. We find that the simulation shows that the proposal works, however, it is tedious.

**Proposal 2:** An alternative mechanism is to let each student submit their segment to a master directory. The files are then merged and the conflict is resolved by any one of the students. To do this each student should have a branch. Since Student D is in charge of the Deliverables which is usually the last part of the report we will assign student the master directory. This is just case, but it does not have to be this way. If another student volunteers or if the team decides on someone else to be in charge of master directory that will work too. The point is that one person is in charge of resolving the merge conflicts at the end.

There is much strength to this mechanism. First, only one student will resolve the merge conflicts and compile the documents. This saves time and effort. Second, the probability of getting extra copies is at a minimum. Third, when it comes to editing it will not be as tedious to recompile and merge again. There are some weaknesses, however. Because the merge conflicts are managed by one student, there is a chance that if an error occurs there it will not be fixed since there is no proofreading mechanism. Finally, a set deadline must be established or else each student can merge multiple versions which would lead to confusion.

To test this we can perform a short simulation. Since I am only one person, what I will do is I will use the poem exercise we did in class and act as each student. The first step is to assign a stanza to each student. For example, A will be stanza 1, B will be stanza 2, C will be stanza 3 and D will stanza 4. I will first act as student D and create a branches for each student. In each branch I act as each student and pull the stanzas from github. I then checkout and go back to being Student D and pull the last stanza. As student D, I merge all the branches and resolve the conflict. I find that the simulation shows that the proposal works.

**Final Recommendation:** My recommendation would go to proposal 2, since it is less tedious and more efficient.

**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 intentional, 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. The first step to answer this question is to define what a tennis game. My model will be based on my perception of the rules of the tennis. I have to admit that I am not entirely clear on the details of the rules of tennis. My basic understanding is depicted in the diagram below:

Based on this understanding, each game contains  $N$  number of sets and each set contains  $J$  number of matches. I will define the question specifically to state Is the game of tennis fair? to Does starting a match first give you an advantage in the greater scheme of one game? If the answer is yes then the game is unfair. If it is no then the model is inconclusive, since it only answers one component of

the game. The reason why I used starting first as the variable to test for is the fact that we do not have any power over the specific endogenous parameters such as players mood, psyche, skill, etc. I will assume that both players are the same in terms of skill level. In terms of exogenous variables like weather, crowd cheering, and other external factors that affect the game, I will ignore those as well for simplicity.

I will assume that starting or serving first gives that player an advantage in the specific match. The players will also alternate who starts first for each match. To model the increased probability of winning a round, I will use an imbalanced coin model. In this game we have Player A and Player B. If Player A starts first the probability of A winning increases and the probability of B winning decreases. The exact numbers will not be determined for the probability and will be determined after experimentation.

Let us focus on Player A. To win, Player A needs to win the majority of sets to win the game. This is done by winning the majority of matches in each set. The number of matches and sets played is determined by the score. If Player A starts every time then Player A would have a greater chance of winning the game. We can use the following equation to determine whether or not a player has an advantage. If the ratio of probabilities of the Player A: PlayerB winning is not equal to 1 then the game is biased.

Ratio=  $P(\text{Player A wins set} \mid \text{starts first}) / P(\text{player B wins set} \mid \text{starts first})$

$P(\text{Player X wins set} \mid \text{start first}) = P(\text{greater side of biased coin})^n$ , where  $n$  is the number of times Player X starts first

Because we are modeling the match using a biased coin, we find that the probability of winning a set for a single player is unaffected if the number of matches is even. This makes sense because Player A and B get the same advantage. However, if the number of matches is odd, then the player that starts first will gain a greater advantage, since that player will start first more times than the other player. The equation will not equal one.

According to the model, tennis is unfair sometimes. The usefulness of the model can be determined by running a Monte Carlo simulation using a computer and a game console system.

**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.

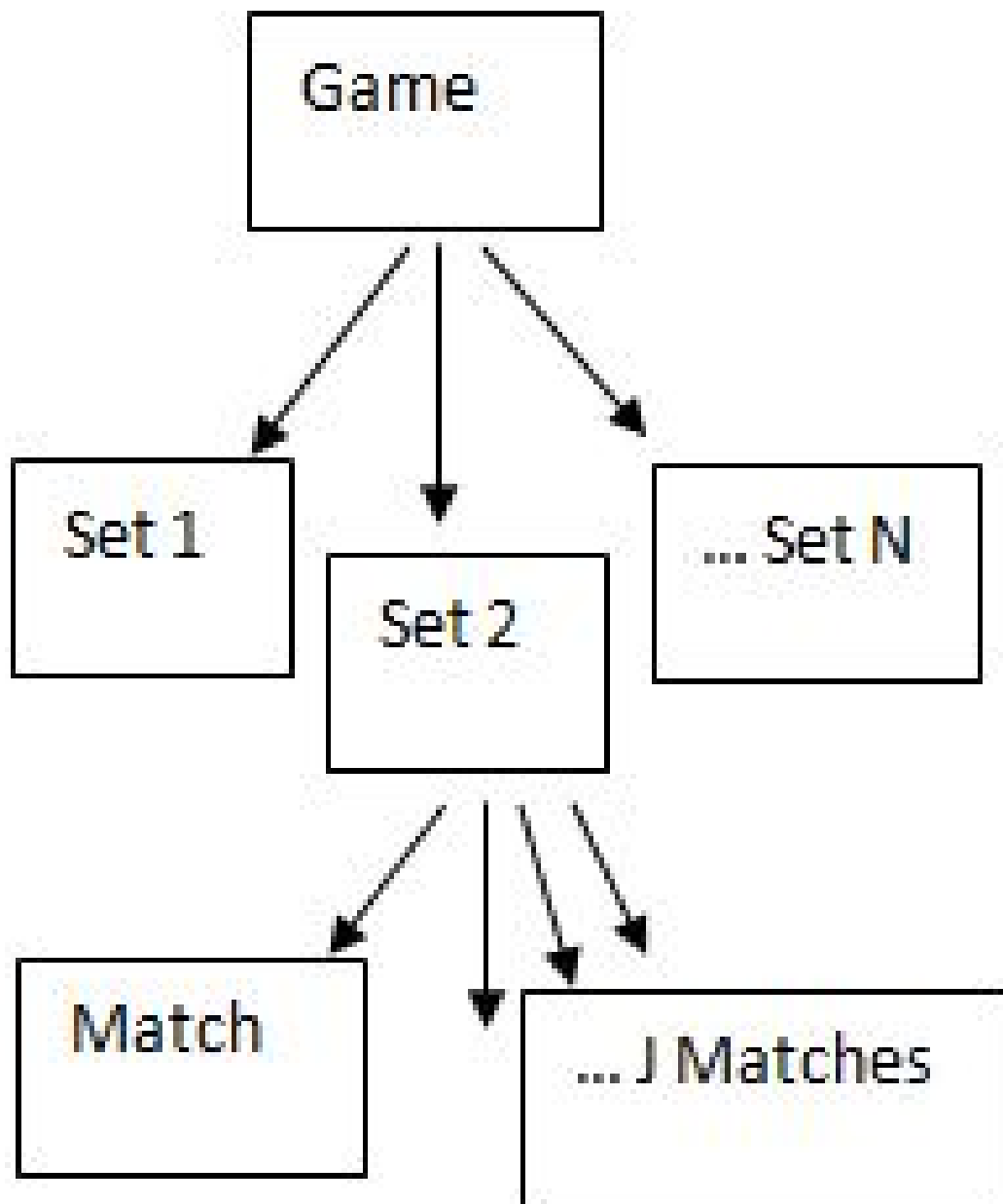


Figure 3: My understanding of a tennis game