AMS 550.400 HW SET 1 Due Date: Oct 8

Last Compiled on October 9, 2012

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:

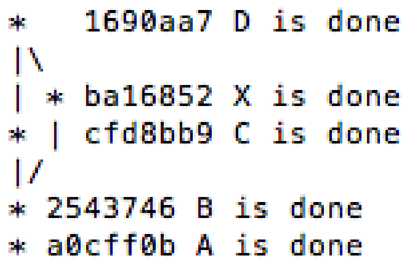
- the *master* branch has commits A, B, C, X and D,
- the *alt* branch has commits A, B, X,

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.

In this problem, I used the following code:

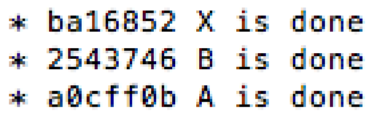
```
cd ~/550400
mkdir ~/550400/honda
cd ~/550400/honda
git init
vi main.txt
git add .
git commit -m ”A is done”
vi main.txt
git add .
git commit -m”B is done”
git checkout -b alt
vi main.txt
git add .
git commit -m”X is done”
git checkout master
vi main.txt
git add .
git commit -m ”C is done”
git merge alt
vi main.txt
vi main.txt
git add .
git commit -m ”D is done”
git log –graph –oneline
git checkout alt
git log –graph –oneline
git push https://github.com/zhendanzhu/honda.git master
git push https://github.com/zhendanzhu/honda.git 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



```
*   1690aa7 D is done
|\
| * ba16852 X is done
* | cfd8bb9 C is done
|/
* 2543746 B is done
* a0cff0b A is done
```

Figure 1: The history graph for master branch



```
* ba16852 X is done
* 2543746 B is done
* a0cff0b A is done
```

Figure 2: The history graph for alt branch

- 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.

```
mkdir newpoem
cd ˜/newpoem
git config –global user.name ”zhendanzhu”
git config –global user.email zhendanzhu@hotmail.com
git remote add stanza1 git://github.com/nhlee/550400.stanza1.git
git remote add stanza2 git://github.com/nhlee/550400.stanza2.git
git remote add stanza3 git://github.com/nhlee/550400.stanza3.git
git init
git checkout master
git pull stanza1 master
vi main.txt
git add .
```

```
git commit -m ” add a title ”
git checkout -b alt1
git pull stanza2
git checkout master
git merge alt1
vi main.txt
git add .
git commit -m ”resolve conflict1”
git checkout -b alt2
git pull stanza3
git checkout master
vi main.txt
git add .
git commit -m ”resolve conflict2”
git remote add origin https://github.com/zhendanzhu/poemmerge.git
git push -u origin master
```

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,

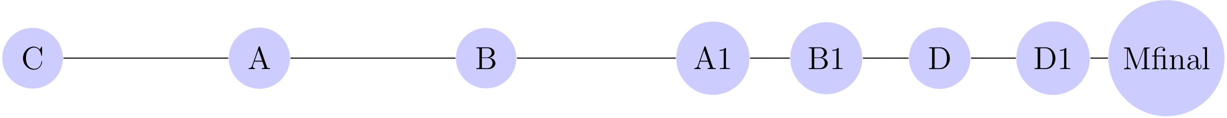
- first, devise a work flow strategy for the team so that they can collaborate asynchronously using `git`,
- next, devise yet another `git` strategy different from your earlier proposal.

Finally,

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

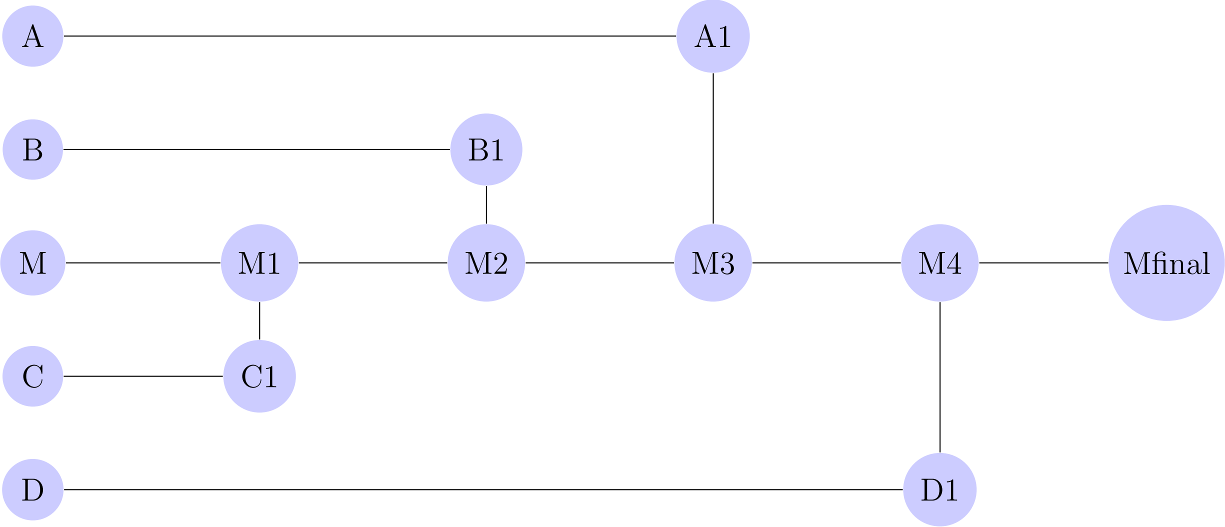
In order to answer this question, *build* a mathematical model, *following* the guideline from IMM. Use Section 1.4 and Section 1.5 of IMM as *role models*. For example, you are to identify which variables are exogenous and which are endogenous. More specifically, among other things, in your model, is 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.*

Problem: Resolve the conflicts between different editions based on workflow design.
When will be the conflicts? Well, if C is behind the others, say haven't finish the timeline while A B D have already start the work. Or they have done the work on different editions, when they merge them together, they will find conflicts in the contents done by other people. The best way to solve it is by keep each teammate informed of the confict so that they can solve it in time. However, given that each person may have flexible schedule, I think the better way is to reduce the total number of merging and marked on the part to be merged.
Outline for the model:
Proposal plan 1: We will have only one branch, i.e, all the work will be done in master branch, and each time one finished his work, he can push it back to the master's branch.



Stength:keep each person's newest edition updated in time.
Weakness: the workflow will be a mess if each one of the team submit the unfinished part. It needs many merges during the process, which will lower the efficiency of the project.

Proposal plan 2: Since AB CD have relatively independent part of the project, each one can work on his/her own branch first, and merge to the master's branch when it is done. Considering different person may work asynchronously. Each time there's a conflict in merging, we can simply keep what it is for our own part and wait for the other one to update their part . Since C is responsible for timeline, it can be done independently in one file. If eventually C finished first, then A B D can follow the timeline to do their part and update their work in a fixed schedule. If C haven't finish his part, then A B D can work in a free way as long as they pull the latest edition from the github before they start working on their part.
Strength: Each one of the team will have its independent part, so that we can reduce the times of merge. Keep completeness of each one's work while make a clear outline for the whole process.
Weakness, the number of conflicts will be a headache when B merged his file with master branch if A, C, D updated their part during B's work. So one person may need to solve the conflicts of merge of several person's editions.



Final Recommendation: Plan 2 will be good, because it has an individual branch for each person. Each one of team can keep track of others work (individual branch) while have a clean clue about the process of the whole project(master branch).

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.

Problem:
Is the game fair?A general standard for this is if the roles of the competitors are reversed, their probability of winning does not change. Our original problem can be breaked down to several parts: whether the player who is first to serve will be in advantage? What will be the chances for the first game server to win the match given the probability of winning rate for each ball? And to what extent will the advantage be?
Outline for the model: We are going to calculate the probability that the first server to win the match. According to tennis rule, one player delivers the ball to start the game, called server; and

one who receives the ball is called receiver. We simplified the rule by stating that each tennis game into a rule that any person who wins two straight points will win the game.And there are 6 games in a match. The possible condition for a game is as following:

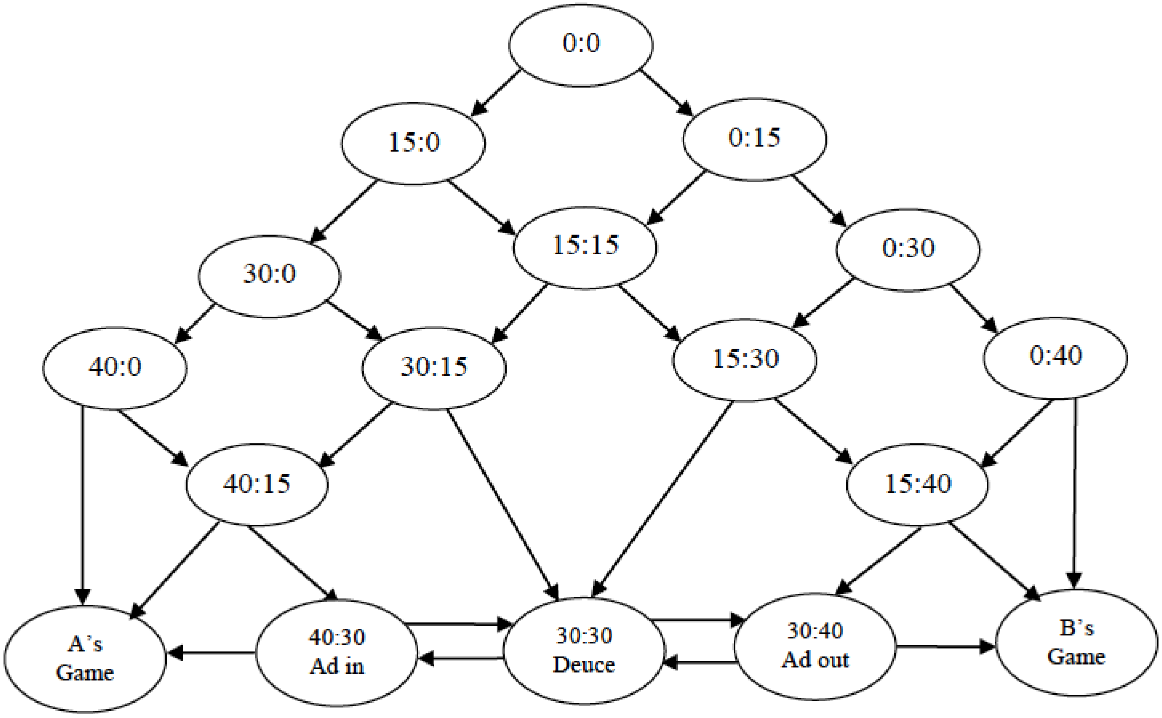


Figure 3: The graph for possible score results

Condition: For both players, the chances for the server to win is P, and the probability for the receiver is 1-P.

Analysis: Whether the game is fair or not depends on the server's winning rate P on each ball. For each player, the chance to win a game is the same, it equals $Q = P^2/(P^2 + (1 - P)^2)$. The chance to lose the game is $(1 - P)^2/(P^2 + (1 - P)^2)$. If P¿1/2, then the winning rate for the server in each game is bigger than 1/2. Given the rule for winning a tennis match is the one who wins the fisrt 6 games will win. The final score can be "6:0", " 6:1" , "6:2" ...The total chance for the first server to win is

$$\sum_{k=0}^{5} \binom{k+5}{6} Q^6 (1 - Q)^k$$

The value for this function is show as following

Code is as following(Figure 4):

We will conclude that the fairness of tennis depends on the capability of each play.The stronger the server, the more advantage he will have in the tennis game. Especially, when P=1/2, the chance for each player to win equals 1/2. The game is absolutely fair. On the other hand, if the receiver is strong enough that the chance for him to win each point exceeds 1/2, then he will have the advantage. So tennis is a fair game if each person has a relatively equal winning rate in both server's game and receiver's game.

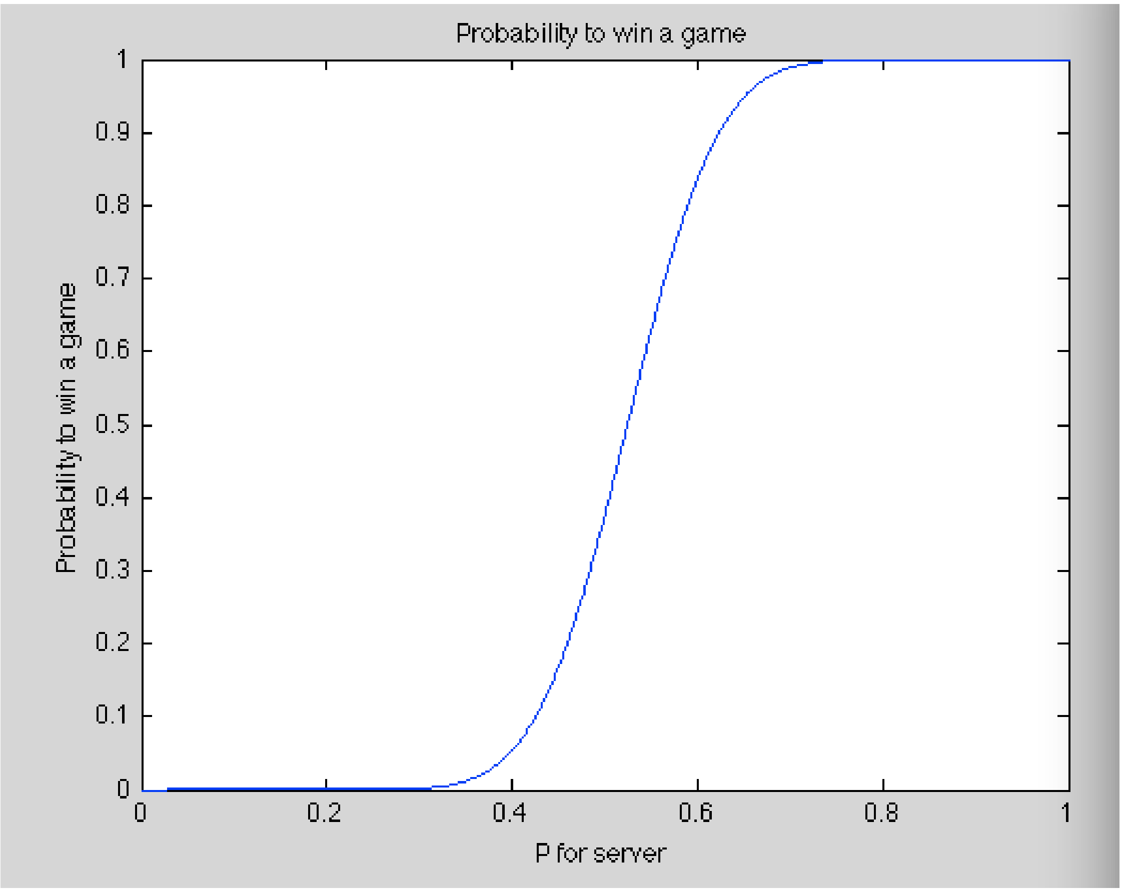
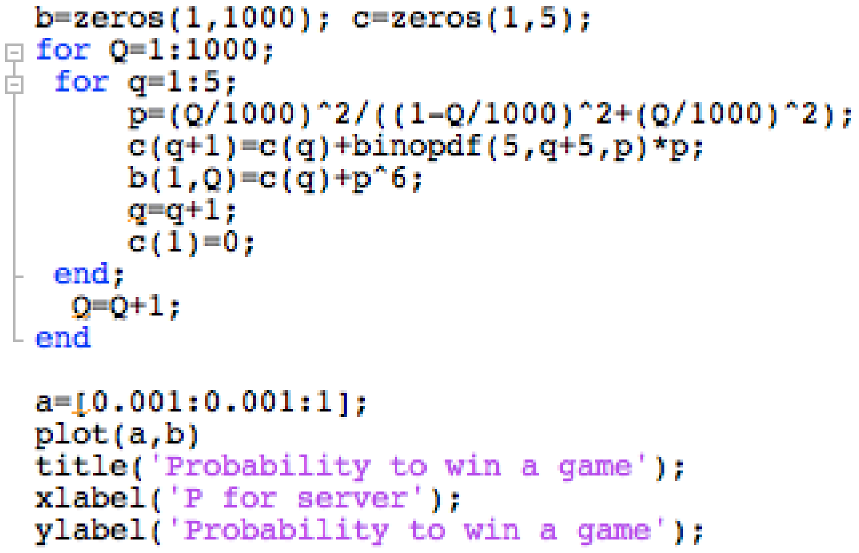


Figure 4: The graph for possible winning rate

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.



```
b=zeros(1,1000); c=zeros(1,5);
for Q=1:1000;
 for q=1:5;
     p=(Q/1000)^2/((1-Q/1000)^2+(Q/1000)^2);
     c(q+1)=c(q)+binopdf(5,q+5,p)*p;
     b(1,Q)=c(q)+p^6;
     q=q+1;
     c(1)=0;
 end;
  Q=Q+1;
end

a=[0.001:0.001:1];
plot(a,b)
title('Probability to win a game');
xlabel('P for server');
ylabel('Probability to win a game');
```

Figure 5: Matlab code