

hw6.sagews

May 5, 2014

Contents

1 Homework 6 - Math 480b - Spring 2014	1
1.1 Problem 1 fork me on github.	1
1.2 Problem 2 git branch	1
1.3 Problem 3 dive into Sage	2
1.4 Problem 4 First Rough Draft of Your Project	2

1 Homework 6 - Math 480b - Spring 2014

Due Friday, May 9, 2014 by 6pm

This homework will be automatically collected from the folder homework6 in your project, sometime after 6pm on Friday, May 9, 2014.

For help email William Stein (wstein@uw.edu) and/or the mailing list sagemath2014@googlegroups.com (You can also try to email Simon Spicer (mlungu@uw.edu), but he is in a far away land)

1.1 Problem 1 fork me on github.

1. Create a fork of the following github repository using your github account: <https://github.com/Math480bSageCourse/gauss>. Put the URL to your fork of this project in your answer below.
2. Create a clone of that fork in the homework6 directory of your project. The result will be that you have a new directory in the homework6 directory of your project with .git subdirectory, etc.
3. Add at least one new function to the class defined in gauss.sage. You could just copy some code from your old homework assignment on this.
4. Commit the new version of your code, and push it to your github repository.
5. Send a pull request to <https://github.com/Math480bSageCourse/gauss>, so after everybody does there homework, <https://github.com/Math480bSageCourse/gauss> will have at least 40 pull requests.

1.2 Problem 2 git branch

1. In your SageMathCloud project, create a branch called crazy of the git repository you created in Problem 1.
2. Checkout that branch and add a crazy buggy function to gauss.sage.

3. Commit your change.
4. Push the resulting new branch to the github account you're using for this course. Once you do this, you should be able to browse the new branch on github with your web browser. Note I did not show you how to push a new branch during class, so you may have to search around to find out how to do this.

1.3 Problem 3 dive into Sage

Choose some function or functionality in Sage, e.g., by browsing <http://www.sagemath.org/doc/reference/> and try to figure out as much as you can about how it is implemented by digging into the source code.

For example, if you choose the `prod` function (you can't choose anything else though), you would start by typing `prod??` into Sage and see that the function is defined in the file `/usr/local/sage/sage-6.2.rc0/src/sage/misc/misc.c.pyx`. You could then type `open /usr/local/sage/sage-6.2.rc0/src/sage/misc/misc.c.pyx` in a terminal to open that file and look around at the source code until you find this line `def prod(x, z=None, Py_ssize_t recursion_cutoff=5):`, and start reading. It may be necessary to open other files. Answer the following questions:

1. What function (or functionality) are you looking at? Briefly describe what it does.
2. Given a working example to illustrate how it works.
3. Is it well documented (in your opinion)?
4. Does the source code make sense to you? If so, what's the basic idea of the algorithm. If not, at what point did you get lost?
5. Functions in Sage tend to fall into three categories:
 - toy: functions that correctly implement some algorithm, but are really slow and naive
 - solid: reasonably good usable functions that implement something and are competitive with other software, etc. Usually a lot of work went into these.
 - world class: Sage beats most of the other attempts out there in commercial or open source software to do the same thing.

Which do you think your function falls under?

1.4 Problem 4 First Rough Draft of Your Project

- Turn in the first rough draft of your project. Put a copy of all files related to your project in the homework6 folder (so I get them when I collect it). Answer the following questions in the blank below, so your random grader will be able to evaluate your project.
- Give a precise description of what your project consists of. Code, paper, etc. E.g., My project is a paper, and I have included a latex document of the current in the directory foo. or My project is to improve the documentation of the following functions in Sage:

This rough draft project will be evaluated by your graders based on effort and overall quality. The graders will:

- completely ignore all typos, rough writing, bugs in code, and anything else technical that would get cleaned up in a polished final draft.

- award points based on whether the project has the potential to be useful or interesting to other people.
- award points based on evidence that significant effort was expended toward creating this rough draft.

(We will make points more precise in the grading guidelines.)