How to review an existing ticket in Sage: (for Sage 8.4)

1. **Review the ticket on trac.sagemath.org:**
   - In your browser, go to trac.sagemath.org/<<ticket number>>
   - review the description of the ticket to see what functionality this ticket addresses
2. **Review the code changes on trac.sagemath.org:**
   - on trac.sagemath.org/<<ticket number>> click on the value next to "branch" (left column of the box describing the ticket) That will open a window showing the changes made by this ticket.
   - review the code changes for correctness. Can you think of any improvements in the efficiency of the code? Any cases the code is failing to address?
   - Make a note of the files and routines that are affected.
3. **Be sure you have no uncommitted changes:**
   - In your terminal, navigate to the the directory where you're doing Sage development.
   - Check if there are uncommitted changes from your work: type `git status`
   - If there are no changes (nothing to commit), go to the next step.
   - Otherwise: type `git stash` to save your changes without a commit
     type `git commit -a` if you want to commit your changes
     This will open a text file. Edit the top line of the file to say something like "<<ticket number>>: <<short message describing your changes>>"
   - Save the changes to that file
4. **Create a branch for testing:**
   - Be sure you are on the master branch: type `git branch` to check this.
   - If you are not on master: type `git checkout master`
   - type `git branch <<branch name>>` to create your branch for testing
   - type `git checkout test` to move to the testing branch
5. **Pull the ticket (this is different from checking out a ticket):**
   - type `git trac pull <<ticket number>>`
6. **Build sage to include the ticket's changes :**
   - In the terminal, be sure you are in the directory where you're doing Sage development and on the test branch. type `./sage -b` ("b" is for "build")
7. **Build the documentation:**
   - In the terminal, be sure you are in the directory where you're doing Sage development and on the test branch. type `./sage -docbuild reference/dynamics html` (of course if the ticket changes a folder other than dynamics, replace that with your folder)
   - The last line of the output from this command will be a local path to a file. Open that file in your browser and check the documentation for the routines changed by the ticket to be sure it looks ok.
8. **Test the documentation:**
   - In the terminal, be sure you are in the directory where you're doing Sage development. type `./sage -t <<path to folder changed by ticket>>` ("t" is for "test")
   - You should test any folder where the ticket may have an effect. For example, you may type: `./sage -t <<path to your Sage folder>>/src/sage/schemes/projective/` to test the whole "projective folder"
   - If your changes also affect other functionality, be sure to test that as well. Make a note of any problems.
   - You may need to test the entire suite of doctests: `make ptestlong` which can take a couple hours
9. **Test the code:**
   - In your terminal, navigate to the affected file(s) and open them in an editor. `cd src/sage/dynamics/<<folder>>/<<file>>`
   - Test cases that should work (lots of them). Test cases that should not work and be sure you get reasonable error messages.
   - Do you best to find fringe cases and do your best to break the code
10. **Modify the ticket in Sage::**
    - In your browser, go to on trac.sagemath.org/<<ticket number>>
    - Add yourself as a reviewer (use your full name, not your trac/github login).
    - Change the status to "positive_review" if all tests were passed and you found no problems while testing.
    - Change the status to "needs_work" if you found any errors at any stage. Be sure to write helpful comments to describe the problems especially the code for an test examples that failed. Sage code can be placed between triple braces: {{{ sage code }}}
    - Then update the ticket.
11. **Clean up::**
    - You probably (but not definitely) want to change back to the master branch and delete the test branch.
    - In the terminal, be sure you are in the directory where you're doing Sage development.
    - type `git checkout master`
    - type `git branch` to make sure that you are on the master branch and to get the name of your test branch
    - type `git branch -D <<branch name>>` to delete your test branch