Program 3 350

(third GitHub assignment… putting instructions on Google Drive so I can edit quickly)

Please e-mail me if you don't see your e-mail in the list and do it from your USC e-mail. Also, include your GitHub Username.

Description:

You will code divide and conquer closest pair as from the book. The algorithm is written at a high level and you need to at least figure out the implementation details before you start coding. Read the project-specific notes carefully. Please read, or at least skim:
Segfault troubleshooting strategies

SSH Instructions:

In Putty/MobaXTerm/WinSCP/Bitvise/… or just plain
SSH to **l-1d43-XY.cse.sc.edu** *on port 222*

- first character is a lowercase L
- ***XY****=01 | 02 | … | 36*

If you're off campus, you will need to have Duo 2 Factor Authentication setup.

From a non-lab Linux machine can
ssh -p 222 <username>@l-1d43-XY.cse.sc.edu

Instructions:

Go to this URL: https://classroom.github.com/a/AADKjE2l (skip the e-mail selection… your GitHub and school e-mail addresses should be associate by now)
Accept the Assignment
Follow the second link to the assignment.
Click on "Clone or download"
Copy the link
(if you're remoting into the Linux labs, login and then switch to your ssh connection here)
<navigate to wherever you want to work>
git clone <the url>
The assignment repositories are private, so you'll need to login.
git config --global --edit and edit the file (you probably did this already)

You can run the scoring function that builds and runs the tests for you and prints the score for you by running the following command (**cd** into the cloned directory first):

python3 score.py

You should see, “**SCORE= 0**” before you get any work done.

You can run the unit tests directly without using the score function (not necessary) with the following command, also from the downloaded directory.

```
cd program-_-<username>     (written generically)
mkdir build && cd build
cmake ..
make
./runUnitTests
```

Do a make clean if it seems to be acting “oddly”.

Code the Divide and Conquer Closest Pair algorithm. Notes:

- The algorithm in the book is a little different, and a little underspecified. Also note that your code needs to return the actual points.
- There is a brute force closest pair algorithm given, closestPairBF(), in metrics.h. Do not change it but feel free to use it for your base case(s), for size two or three subproblems. Do not change metrics.h. It will be replaced. You may use dist() as well.
- Do not *repeatedly* sort the input or subproblems (it will go **$O(n \log^2 n)$**)
- Since this is one monolithic problem, there will be partial credit given for non-working code (segfaults in particular)... max score is 60.
- Code that does not *compile* will not be graded (it gets a 0).

Committing (backup and submission)

When I’m done or want to backup (**git** is for version control) I just

```
git add .
git commit -m “<some message>” // “final submission” makes sense if you’re done
git push origin master
```

We will grade the last submission up to the deadline (the very last submission until we won’t accept it anymore, possibly with a late penalty).

No need to submit code to dropbox.