Scientific Computing — Fitchburg State University
Department of Mathematics
HW #6 — October 21, 2016

Scientific Computing

Fitchburg State University
Department of Mathematics
HW #6

October 21, 2016

The following problems should be done in julia. Type up the code you used to find your results and answer all questions. Submit your homework via Blackboard by uploading the ipynb file. This assignment should be submitted no later than 11:59pm on October 28, 2016.

1. This problem take the types and functions from Problem #3 in HW #5 and place them in a module called Geometry.
   (a) Write the module and you should export the following: `Point, Polygon` types and the functions `distance, perimeter, plot` and if you wrote an `area` function. You should put this in a separate file.
   (b) Write a `is_rectangle` function that tests if it a rectangle. The order of the points is important. For example, (0,0), (2,0), (2,1), (0,1) is a rectangle, but (0,0), (2,1), (0,1),(2,0) is not. Hint: to check if you get a right angle, create a vector from the points and use the dot product. Include this in your module.
   (c) (Extra Credit) Consider writing additional constructors for Point and Polygon that take strings and parses the strings. You may need to use regular expressions. Check out the Julia documentation on Strings.
2. Write a Unit test for your Geometry module in Problem #1 and run it to make sure it passes all test. Specifically, you should have the following:
   (a) Write an `isequal` function that tests if two points are equal.
   (b) See if creating a point is actually a `Point` type. Hint: use `isa`.
   (c) See if creating a polygon is actually a `Polygon` type.
   (d) Test the `distance` function.
   (e) Test the `perimeter` function.
   (f) If you wrote an area function, test the `area` function.
   (g) If you wrote other constructor functions, test them.
3. This problem is about timing results again and running some parallel code. See the code in class and visit the Julia manual on Parallel computing.
   (a) Time your results to simulate 2,000,000 draws and testing for a flush (or one of the other hands, if you didn't get flush to work)
   (b) Restart julia with 2 processes in the command line. Spawn two processes and test 1,000,000 draws and test for a flush for each process. Time your results. If you have more than 2 cores, you can try to increase to 4 or 8 cores and rerun.
   (c) Time the code in part (b) and comment on the results. Is it what you expect?
   (d) Rewrite the simulation to use the parallel for loop using more than 1 process. Time your results.
   (e) Discuss your results. in (d).