COMS 3203 Discrete Mathematics Fall 2018

Amazing Prime Numbers Project

Due Monday, November 26th

Please submit your own original work. Project assignments will be checked with robust plagiarism detection software used on all your submissions and code found online. All code has to be done in Python. Please note, this is the first time we offer this assignment. We may post modifications/clarifications to this homework as we hear your feedback.
Be creative and have fun!

The following assignment is about divisibility and prime numbers. Every section requires that you write some python code and report your results. Besides submitting one python file, please submit a report that answers the questions below along with your observations and findings. In one paragraph, please explain in your report how you shared the workload in your team.

1 Euclid's algorithm

Euclid Algorithm was invented by the Greek mathematician Euclid about 2,300 years ago. The algorithm calculates the greatest common divisor/factor (GCD) of two integers. (Please refer to your lecture for more details about how and why it works).

1. Write a python function that implements the Euclid's algorithm to calculate the GCD of two integers.
2. Now use your algorithm to calculate the gcd of several pairs, for example, GCD(4278, 8602), GCD(406, 555), and GCD(244, 354). Try large numbers. Print all the steps taken by your algorithm and add that to your report.
3. Report all your results in a table to show how the Euclid algorithm scales as we increase the number of digits (Be ambitious in the number of digits).

2 How high can you go in generating prime numbers?

We would like to study (later in this assignment) the distribution of primes. For this, we need to generate a very large number of primes.

1. Write a python function that generates primes up to a certain n using the Sieve of Eratosthenes.
2. Report all your results in a table to show how the sieve scales as we increase the number of digits (Be ambitious in the number of digits).

3 Primality test

Implement three primality test functions using the two following approaches:

1. Trial division: For an input n, check if there is a prime number between 2 and $\sqrt{n}$ that divides n.
2. Using Sieve of Eratosthenes.
3. Fermat little theorem **BONUS**.
4. Report all your results in a table to show how each method scales as we increase the number of digits (Be ambitious in the number of digits).

4 Prime factorization

Implement a fast prime factorization algorithm using two ways:

1. Trial division.
2. Another faster method (example, the Pollard-Strassen method) **BONUS**.
3. Report all your results in a table to show how each method scales as we increase the number of digits (Be ambitious in the number of digits).

5 Prime distribution

Now, we are interested in calculating few statistics about prime number distribution.

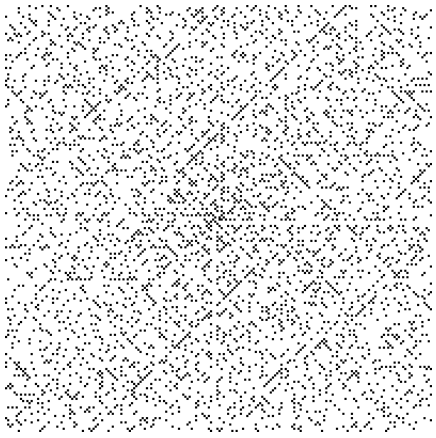
1. Collect a prime number dataset, that is the list of prime numbers. Your dataset should have at least one million prime numbers. You can either generate it with the functions you created above or download it.
2. Calculate the proportion of primes that ends with 1 (1 is the right digit)? with 3? with 7? with 9?
3. Collect the following statistics in a table:
   (a) Calculate the proportion of primes ending with 1 that are followed by primes ending with 1? with 3? with 7? with 9?
   (b) Calculate the proportion of primes ending with 3 that are followed by primes ending with 1? with 3? with 7? with 9?
   (c) Calculate the proportion of primes ending with 7 that are followed by primes ending with 1? with 3? with 7? with 9?
   (d) Calculate the proportion of primes ending with 9 that are followed by primes ending with 1? with 3? with 7? with 9?
4. How many twin primes are there?
5. Let $\pi(x)$ representing the number of primes less than x. Using the data you collected, plot $\pi(x)$.

Report all your observations in your report.

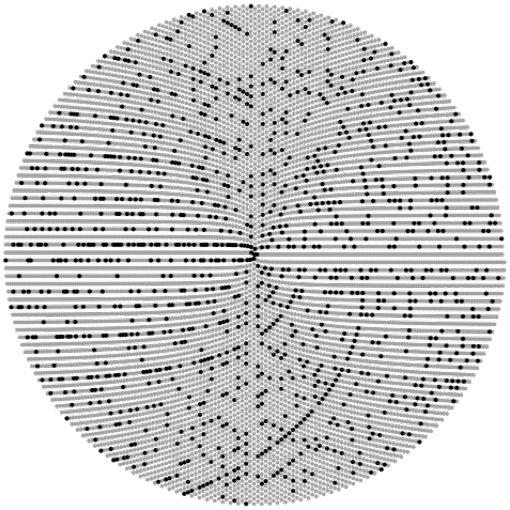
6 Prime visualization BONUS

Now, we are interested in visualizing prime numbers. Visualizing prime numbers has been a topic of interest for a long time and there were several attempts to visualize as a spiral called Ulam Spiral, and other variants.

Write a Python function that creates a compelling visualization of prime numbers. You can be inspired by existing prime visualizations but we encourage you to be creative and come up with your own. Here are few examples from the literature.



By Grontesca at the English Wikipedia, CC BY-SA 3.0
https://commons.wikimedia.org/w/index.php?curid=1924394



By Claudio Rocchini - Own work, thanks to www.numberspiral.com
CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=2212912

7 Wrap it up in a Menu

Now, create a menu prompting the user to use any of your functions.