PRIMALITY TESTING

INITIAL IMPLEMENTATION PLAN

Team –
Arijit Banerjee
Suchit Maindola
Srikanth Manikarnike

Problem Statement

The aim of this project is to implement the widely studied problem – Primality testing – with as fast a running time as possible. In saying so, our final goal is to successfully be able to implement a breakthrough achieved in 2002 by Agrawal, Kayal and Saxena (AKS)[1][8] who showed a polynomial time deterministic algorithm for determining whether a given number is prime. Although, this is a fast algorithm that works on all inputs and does not have assumptions and is not randomized, it does not have the best running time. Some algorithms that run faster than AKS are the Miller-Rabin algorithm and the Lucas-Lehmer test for Mersenne numbers. These algorithms, however, are deterministic only for certain subsets of inputs. The Miller-Rabin algorithm is fully deterministic and runs in polynomial time over all inputs, but its correctness depends on the truth of the yet-unproven Generalized Reimann Hypothesis (GRH) [2]. The Lucas-Lehmer test for Mersenne numbers, as the name suggests works only for Mersenne numbers. Hence, as we try to achieve our main goal, we also wish to explore these alternative methods of testing primality.

Literature Review

A natural number is said to a prime number if it is greater than one and has no other advisors other than 1 and itself [3]. The problem of testing the Primality of a given number has existed for 2300 years. Numerous approaches have been tried till date to determine if a given number is prime. Of these, some naïve techniques include – Sieve of Eratosthenes, Pascal's triangle, Winston's theorem [4]. Prime numbers are used extensively. Public key cryptography, modular arithmetic, pattern recognition are a few examples.

After substantial literature study, we have decided to implement the following three methods

1. AKS Algorithm
   Salembier and Southerington [5] describe optimizations for implementing AKS Primality Test using LiDLA library in C++. They use dgcd() and bgcd() functions for Euclidian division and binary classification respectively. We are planning to use a similar approach for our implementation and later suggest optimizations if time permits.

2. Lucas-Lehmer test for Mersenne numbers
   Crandall and Pomerance [6], in their book “Prime Numbers: A Computational Perspective” provide information on implementing this algorithm. This will be our second strategy, which will be limited only to Mersenne numbers.

3. Miller-Rabin algorithm
   This algorithm is fully deterministic and runs in polynomial time over all inputs, but its correctness depends on the truth of the yet-unproven generalized Reimann hypothesis [7]. Implementation of this algorithm is fairly easier than the previous two strategies.

Our approach

We plan to implement all three approaches. Implementing strategy-3 is fairly complicated and would require significant time and effort as compared to strategies 1 and 2. Since the Lucas-Lehmer and Miller-Rabin tests work better than the AKS algorithm for certain sets of inputs, we believe that if we can determine whether an input is a Mersenne number or if GRH is true for it, in deterministic polynomial time, we can decide which of the three strategies to apply to our input. This, however, is still conjecture and an approach. We believe we need to explore this idea further for it to improve the efficiency of our primality tester. Any and all inputs on whether this could prove to be a viable option are welcome.

Timeline

As of November 1, 3011, we plan to

- Explore our approach to compute the complexity in figuring out whether the input is a Mersenne number and GRH is true for it.
- Implement all three of our strategies

References

[1] – Agrawal, Kayal and Saxena. Primes is in p. *Annals of Mathematics* (2004), 781—793.

[2] – Wikipedia. AKS Primality Test. http://en.wikipedia.org/wiki/AKS_primality_test#Importance.

[3] – Wikipedia. Prime number. http://en.wikipedia.org/wiki/Prime_number.

[4] – Scott Aaronson. The Prime Facts: From Euclid to AKS. 2003

[5] – Salembier and Southerington. An Implementation of AKS Primality Test. 2005.

[6] – Crandall and Pomerance. *Prime Numbers: A Computational Perspective*. Springer Verlag. 2005.

[7] – Wikipedia. Miller-Rabin Primality Testing. http://en.wikipedia.org/wiki/Miller_Rabin_primality_test#Deterministic_variants_of_the_test

[8] – Advanced Algorithms, University of Utah, Fall 2011. https://learn-uu.uen.org/courses/48426/files/3870539/download?wrap=1.