Abstract

A) Purpose of the Experiment

Modern cryptography relies on probabilistic primality tests to generate large primes. Such tests result in false positives called pseudoprimes. It is an open problem to find deterministic primality tests that are efficient for large numbers. My research focuses on the effectiveness of different primality tests by analyzing the sizes of their smallest pseudoprimes, the number of pseudoprimes up to a certain maximum value, and their pseudoprimes after restricting to k-rough numbers.

B) Procedure

I analyzed primality tests based on Lucas-Type numbers with different parameters. A special subcase is Fermat's well-known primality test. I ran computations to create visualizations of the smallest pseudoprimes and the number of pseudoprimes. I determined the cases which stand out from these graphs.

C) Data

The least promising case I found by looking at the number of pseudoprimes is the Lucas-type sequence (5, 2), which has 53 pseudoprimes up to 1,000. However, I found that this number can be reduced to 0 by restricting to 3-rough numbers. The best case I found is Lucas-type sequence (8, 6), which only has 4 pseudoprimes up to 1,000. It still has one pseudoprime after being restricted with 3-rough. This showcases that one primality test that initially looks worse than the other might actually be a better primality test after excluding pseudoprimes that are divisible by 2 or 3. I then generated more general graphs to compare primality tests after restricting to 3-rough pseudoprimes.

D) Conclusions

This data shows that we can substantially improve the efficiency and the quality of many of the primality tests this way.