1 Primality Testing

First, we will treat the various methods of determining whether or not a number is prime, or, as is usually easier, determining whether a number is composite. At a first glance, it seems strange to have methods for such a thing that are distinct from factorization methods - after all, if you can obtain the prime factorization of a number, it is trivial to determine primality from that factorization. However, factorization methods are in general very computationally expensive compared to some of the methods we will examine in this section.

1.1 Primality Tests and Composite Tests

Any viable computational method for testing primality explicitly is composed of a condition on a number n that, when met, necessitates that n is prime. Thus if the condition is not met, n is composite. Such tests are certainly extremely convenient to directly determine the primality of n, but unfortunately methods of this form are usually a combination of very complex and restricted to n of a particular form or within some bounded range.

In addition to these primality tests, which never fail on determining primality, we also have many tests that are comparatively simple and computationally efficent, which occasionally fail to identify primes, but never indicate that a composite number is prime. We will call these tests, which never fail on determining compositeness, compositeness tests.

It is of the utmost importance to note, before moving on, that while the conditions of a primality or compositeness test being met guarantees that the number is either prime or composite respectively, and a failed primality test proves compositeness, a failed compositeness test does not necessitate that the number is prime.

1.2 The Sieve of Eratosthenes

The Sieve of Eratosthenes is among the first algorithmic methods for factorization and primality testing. Admittedly, it is very crude, but the theoretical basis of the method gives us some of the fundamental ideas for developing further primality tests. Additionally, the Sieve of Eratosthenes also can be used for algorithmic factorization of a number - as such, we will treat it rather lightly here and revisit it in more detail when we examine factorization methods.

The core idea of the Eratosthenes' sieve is as follows:

1.3 Fermat's Theorem and Resulting Methods

1.3.1 Fermat's Theorem

1.3.2 Pseudoprimes and Carmichael Numbers

1.3.3 Computational Viability of Fermat's Theorem Methods

1.3.4 Improvements due to Eulers Criterion

1.3.5 Euler Pseudoprimes

1.3.6 Strong Pseudoprimes

1.3.7 Implementation of a Strong Pseudoprime Test

1.3.8 Remarks and Numerical Data