Probabilistic Primality Testing

Probabilistic primality testing is a class of algorithms used to determine whether a given number is a prime number or a composite number. Unlike deterministic algorithms that always provide a correct answer, probabilistic algorithms may occasionally produce an incorrect result, but the probability of error is extremely low.

One well-known probabilistic primality testing algorithm is the **Miller-Rabin algorithm.** The Miller-Rabin algorithm is based on the properties of certain types of numbers called "strong pseudoprimes." A strong pseudoprime is a composite number that behaves like a prime number under specific conditions.

Here's a high-level overview of the Miller-Rabin algorithm:

Input: An odd integer n to be tested for primality and a parameter k, which determines the accuracy of the test.

Write n as 2^r * d + 1 where r is the largest integer such that 2^r divides n-1 and d is an odd integer.

Witness Loop: Repeat the following k times:

a. Choose a random integer a from the range [2, n-2].

b. Compute x = a^d mod n.

c. If x is not congruent to 1 (mod n) and x is not congruent to -1 (mod n), then continue to the next iteration of the loop.

d. Repeat r times or until x is congruent to -1 (mod n): x = x^2 mod n.

e. If x is not congruent to -1 (mod n) after r iterations, then n is definitely composite. Otherwise, continue to the next iteration of the loop.

If the algorithm has not identified n as composite after k iterations, then n is considered probably prime.

The accuracy of the Miller-Rabin algorithm depends on the chosen value of k. Larger values of k result in a lower probability of error but require more computational effort.