Sieve of Eratosthenes

Khritish Kumar Behera

November 13, 2024

Introduction

The Sieve of Eratosthenes is a highly efficient algorithm for finding all prime numbers up to a given integer n. It operates on the principle of marking off multiples of prime numbers, leaving only the primes unmarked. This report delves into the implementation and time complexity of the Sieve of Eratosthenes algorithm.

Algorithm

1. Initialization:
    - Create a boolean array prime[0...n] and initialize all entries as True.
    - Mark prime[0] and prime[1] as False as 0 and 1 are not prime numbers.
2. Sieving Process:
    - Iterate over all numbers from 2 to the square root of n:
        - If prime[p] is still True:
            - Mark all multiples of p as False from p^2 to n in increments of p.
3. Output:
    - Print all indices p for which prime[p] is True. These are the prime numbers up to n.

Psudocode

```
SIEVE(n):
  primes = [True] * (n + 1)
  For (p = 2 to sqrt(n)) // both inclusive
  {
    if primes[p] == False:
      continue
    For (j = p*p to n with step size of p)
    {
      primes[i] = False
    }
  }
  return primes
```

Time Complexity

The time complexity of the Sieve of Eratosthenes is $O(n \cdot log(log(n)))$. This is because the outer loop iterates from 2 to the square root of n, and the inner loop marks off multiples of a prime number. The inner loop's iterations are dominated by the harmonic series, which is approximately $log(log(n))$.