**Divisibility Operations and Primes**

**Document Id 31**

Perhaps one of the most important intersections between mathematics and

computer science is the use of primes

* Prime test

if number a is a divisor of n, then n/a is also a divisor

a or n/a <= root(n)

If a mod n = 0 then n can not be prime

11 mod 3 = 2 7 mod 3 =1 6 mod 2 = 0

* Prime test

Fermat's test for composites

https://en.wikipedia.org/wiki/Primality\_test#Fermat\_primality\_test

Does not always work for the contrapositive

* Probabilistic tests

https://en.wikipedia.org/wiki/Primality\_test#Probabilistic\_tests

* Miller–Rabin uses Fermat's test and a probability argument

<https://en.wikipedia.org/wiki/Primality_test#Miller.E2.80.93Rabin_and_Solovay.E2.80.93Strassen_primality_test>

**The sieve of Eratosthenes**

Prime test

Very inefficient

**for (int i=2; i<n; i++)  
 if (n%i==0) return false;  
  
return true;**

Increase efficiency

* if number a is a divisor of n, then n/a is also a divisor

a or n/a <= root(n)

**for (int i=2; i< root(n); i++)  
 if (n%i==0) return false;  
  
return true;**

* To the primes between 1 and a million => 100000 iteration very inefficient
* The sieve of Eratosthenes is mujch more effcient
* In some situations is useful to be able to generate a sequence of prime numbers

<https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes>

* It begins by assuming that all numbers are prime then takes the first prime number

and removes all of its multiples

**1 generate a list of counting numbers**

**2 assume that all numbers are prime**

**3 take the first number in the list and removes all of its multiples in the list**