Number Theory

This document covers

Definitions

Factors, Divisibility and modulo arithmetic

Fundamental Theorem of Arithmetic

HCF/LCM

Euclids Algorithm

Floor Ceiling Functions

## Factors, divisibility and Modulo arithmetic

## Definitions

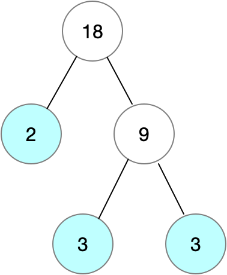
|  |  |
| --- | --- |
| Divisor | If we say p is a factor or divisor of q and q is divisible by p. P is a multiple of q |
| Fundamental theorum of Arithmetic | Any integer can be expressed as the product of prime factors |
| Highest Common Factor | The highest number that is a divisor of two number. Given two integers , and we can calculate the highest common factor as |
| Lowest Common Multiple | The lowest number which is a multiple of two numbers |
| Relating HCM and LCM |  |
| Euclids Algorithm |  |
| Floor |  |
| Ceiling |  |

## Fundamental Theorem of Arithmetic

Any integer is either prime itself prime or can be expressed as a product of prime factors

Where are successive primes and are powers of that prime. For any given p, the corresponding can be zero. We can find the prime factors of any given number by continually dividing through. The following shows how to extract the prime factors of 18

Figure Prime Factorisation of 18



### Testing for Primes

#### Naïve Implementation

The following naïve implementation is

public static bool IsPrimeNaive(int n)

{

if (n <= 1) return false;

for (int i = 2; i < n; i++)

{

if (n % i ==0)

return false;

}

return true;

}

#### Simple Optimisation

We can optimise our naïve algorithm by observing that we only need to test factors up to for the simple reason that any factors greater than must have a corresponding factor less than that we will have already tested by the time we get to

// Question: Write a is prime with square root optimisation

public bool IsPrimeUsingSquareRoot(int n)

{

if (n < 2)

return false;

if (n == 2)

return true;

// The definition of a prime is an integer x

// which is not exactly divisible by any

// number other than itself and one. If a

// number x is not prime it can be written as

// the product of two factors a x b. If both

// a and b were greater than the square root of

// x then a x b would also be greater than x and hence

// a x b is not x. SO testing all factors up to floor(root(x))

// is sufficient as if one factor is floor(root(x)) the other factor must

// be less than that

// hence test the n-2 integers from

// 2,..., Floor(Root(N))

return Enumerable.Range(2, (int)Math.Floor(Math.Sqrt(n)))

.All(i => n % i > 0);

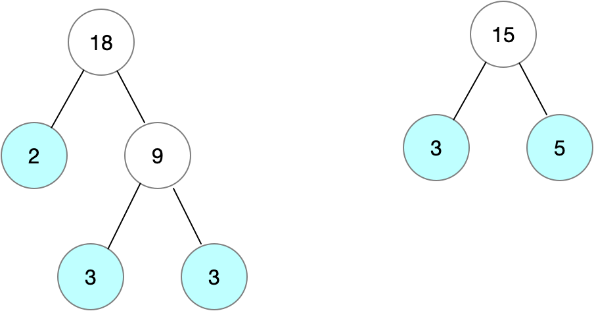
}

## HCF/LCM

### Highest Common Factor (HCF)

Given two integers x and y and their corresponding prime factorisations

We can calculate the highest common factor as

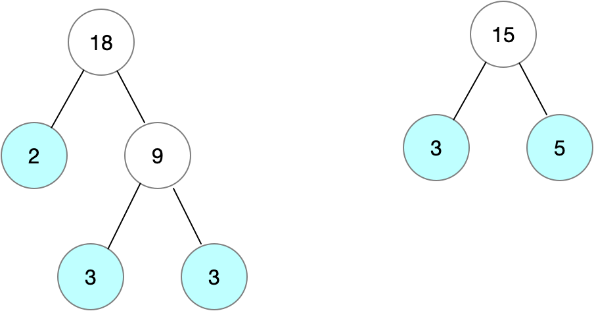


,

### Lowest Common Multiple (LCM)

Given two integers x and y and their corresponding prime factorisations

We can calculate the lowest common multiple as



### Relating HCF and LCM

Given two integers x and y and their corresponding prime factorisations

We can show there is a relationship between lcm and hcf.

So we now know that

This is very powerful as we have efficient algorithms for calculating the hcf, whereas we do not have efficient algorithms for carrying out prime factorisation.

## Euclid’s Algorithm for GCD

#### Proof

##### Show that gcd(a,b) is a divisor of a-b

By the definition of a divisor we know that

##### Show that gcd(a,b) is a common divisor of b and a-b

In the previous step we showed that gcd(a,b) is a divisor of a-b and by definition gcd(a,b) is a divisor of b. We hence know that gcd(a,b) is a common divisor of a and a-b. We know that gcd(a,b) must be less than or equal to gcd(b,a-b) by the definition of gcd(b,a-b) as the **greatest** common divisor

##### Show that gcd(b,a-b) is a divisor of a

By the definition of a divisor we know that

##### Show that gcd(b,a-b) is a common divisor of a and b

By definition gcd(b,a-b) is a divisor of b and we have shown that gcd(b,a-b) is a divisor of a. So we know that gcd(b,a-b) is a common divisor of a and b. Because gcd(a,b) is the **greatest** common divisor of a and bwe know that

Taken (4) and (9) together we have shown that

##### Show that gcd(b,a-b)=gcd(b,a%b)

**We have shown that** . We can apply the formula multiple times

The definition of the % operator is

Letting and substituting into the right hand side of (10) we have

We have now proved Euclids algorithm that

#### Implementation (C#)



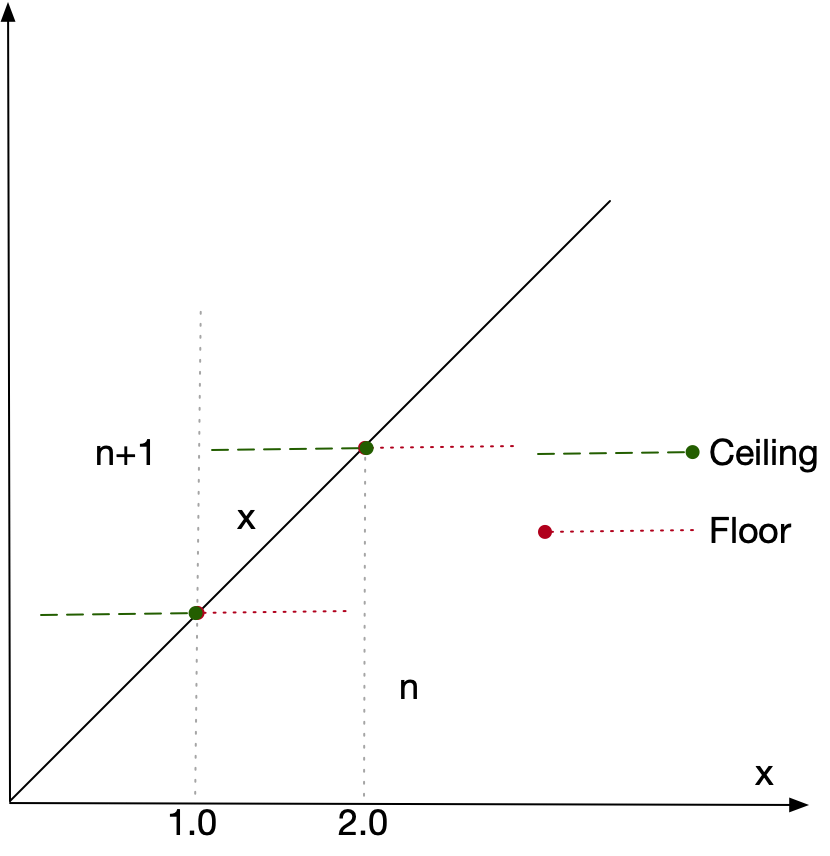
## Floor/Ceiling Functions

### Definitions

Floor – The greatest integer less than x

Ceiling – The smallest integer less than x

Figure Floor/Ceiling



Listing Examples



Fractional Part

The floor gives the integer part of a real and subtracting the floor from the real gives the fractional part

We can use this notation to calculate possible values of

### Property Summary

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | , if n is an integer and x a real |
|  | if n is an integer and x a real |
|  | , if n is an integer and x a real |
|  | if n is an integer and x a real |
|  | if f is continuous, monotonically increasing with the property that if then |

### Property Detail

#### Property 7

#### 

#### Property 8

#### 

#### Property 16

If we define function

where is **continuous** and **monotonically increasing** and where has the following special property

**Property P:** if then

Then for all for which the property P holds

#### Proof

In the simple case where we have nothing to do. We hence focus on the case where .

From the definition of the ceiling function

Because is monotonically increasing

Because ceiling is non decreasing

Assume

*Because ceiling only deals in integers*

This means the monotonically increasing function f must increase above in order to make it possible for This means that the following two things must be true

The special property P means that y must be an integer as is by definition an integer. But there cannot be an integer between x and so we have a contradiction and hence it is not possible for

and hence