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 $p\|q$ 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 $x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$
Highest Common Factor	The highest number that is a divisor of two number. Given two integers $x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$, and $y = p_1^{b_1} p_2^{b_2} \dots p_n^{n}$ we can calculate the highest common factor as $hcm(x, y) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \dots p_\infty^{min(a_n,b_n)}$
Lowest Common Multiple	The lowest number which is a multiple of two numbers
Relating HCM and LCM	$lcm(x, y) = \frac{x \times y}{hcf(x, y)}$
Euclids Algorithm	$\gcd(a, b) = gcd(b, a\%b)\#(10)$
Floor	$floor : \mathcal{R} \longrightarrow \mathbb{Z}$ $floor(x) = \lfloor x \rfloor = max\{a \in \mathbb{Z} \| \ a \leq x\}$
Ceiling	$ceiling : \mathcal{R} \longrightarrow \mathbb{Z}$ $ceiling(x) = \lceil x \rceil = max\{a \in \mathbb{Z} \| \ a \leq x\}$

Fundamental Theorem of Arithmetic

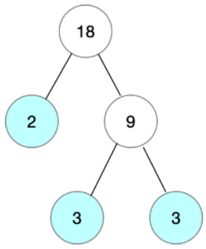
Any integer is either prime itself prime or can be expressed as a product of prime factors

$$x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

Where $p_1 \dots p_n$ are successive primes and $a_1 \dots a_n$ are powers of that prime. For any given p, the corresponding a 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

$$18 = 2^1 \times 3^2$$

Figure 1 Prime Factorisation of 18



Testing for Primes

Naïve Implementation

The following naïve implementation is $O(n)$

```
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 $\sqrt{n}$ for the simple reason that any factors greater than $\sqrt{n}$ must have a corresponding factor less than $\sqrt{n}$ that we will have already tested by the time we get to $\sqrt{n}$

```
// 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)

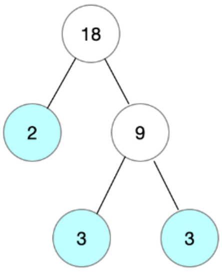
Given two integers x and y and their corresponding prime factorisations

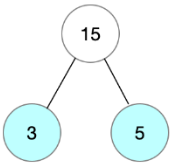
$$x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

$$y = p_1^{b_1} p_2^{b_2} \dots p_n^{n}$$

We can calculate the highest common factor as

$$hcm(x, y) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \dots p_\infty^{min(a_n,b_n)}$$





$18 = 2^1 \times 3^2$, $15 = 2^0 \times 3^1 \times 3^5$

$hcf(15,18) = 2^{\min(0,1)} \times 3^{\min(1,2)} \times 5^{\min(0,1)} = 3$

Lowest Common Multiple (LCM)

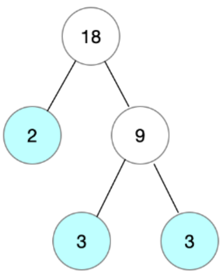
Given two integers x and y and their corresponding prime factorisations

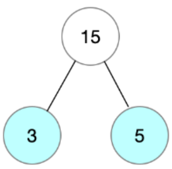
$$x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

$$y = p_1^{b_1} p_2^{b_2} \dots p_n^{n}$$

We can calculate the lowest common multiple as

$$lcm(x, y) = p_1^{max(a_1,b_1)} p_2^{max(a_2,b_2)} \dots p_\infty^{max(a_n,b_n)}$$





$18 = 2^1 \times 3^2$

$15 = 2^0 \times 3^1 \times 3^5$

$lcm(15,18) = 2^{\max(0,1)} \times 3^{\max(1,2)} \times 5^{\max(0,1)} = 2 \times 3^2 \times 5^1 = 90$

Relating HCF and LCM

Given two integers x and y and their corresponding prime factorisations

$$x = p_1^{a_1} p_2^{a_2} \dots p_n^{a_n}$$

$$y = p_1^{b_1} p_2^{b_2} \dots p_n^{n}$$

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

$lcm(x,y) = p_1^{max(a_1,b_1)} p_2^{max(a_2,b_2)} \dots p_\infty^{max(a_n,b_n)}$

$hcf(x,y) = p_1^{min(a_1,b_1)} p_2^{min(a_2,b_2)} \dots p_\infty^{min(a_n,b_n)}$

$hcf(x,y) \times lcm(x,y) = p_1^{min(a_1,b_1) \times max(a_1,b_1)} p_2^{min(a_2,b_2) \times max(a_1,b_1)} \dots p_\infty^{min(a_n,b_n) \times max(a_n,b_n)}$

$hcf(x,y) \times lcm(x,y) = p_1^{a_1 \times b_1} p_2^{a_2 \times b_2} \dots p_n^{a_n \times b_n} = x \times y$

So we now know that

$$lcm(x,y) = \frac{x \times y}{hcf(x,y)}$$

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

$$\mathrm{a} = \mathrm{x} \times \mathrm{gcd}(a, b) \tag{1}$$

$$b = y \times \mathrm{gcd}(a, b) \tag{2}$$

$$\mathrm{a} - \mathrm{b} = (\mathrm{x} - \mathrm{y}) \times \mathrm{gcd}(a, b) \tag{3}$$

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

$$\mathrm{gcd}(a, b) \leq gcd(b, a - b) \tag{4}$$

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

By the definition of a divisor we know that

$$\mathrm{a} - \mathrm{b} = \mathrm{m} \times \mathrm{gcd}(b, a - b) \tag{5}$$

$$\mathrm{b} = \mathrm{n} \times \mathrm{gcd}(b, a - b) \tag{6}$$

$$\mathrm{a} = (m + n) \times \mathrm{gcd}(b, a - b) \tag{8}$$

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 b we know that

$$\mathrm{gcd}(a, b) \geq gcd(b, a - b) \tag{9}$$

Taken (4) and (9) together we have shown that $\mathrm{gcd}(a, b) = \mathrm{gcd}\ (b, a - b)$

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

We have shown that $\gcd(a,b) = gcd(b, a-b) = \gcd\,(a-b, b)$. We can apply the formula multiple times

$$\gcd(a,b) = \gcd(a-b,b) = \gcd(a-2b,b) = \gcd(a-qb,b) \tag{10}$$

The definition of the % operator is

$$\boldsymbol{a\%b = a - \left(\frac{a}{b}\right) \times b} \tag{11}$$

Letting $q = \frac{a}{b}$ and substituting into the right hand side of (10) we have

$$\gcd(a,b) = \gcd(a-b,b) = \gcd(a-2b,b) = \gcd(a\%b,b) = \gcd(b,a\%b) \tag{10}$$

We have now proved Euclids algorithm that

$$\gcd(a,b) = gcd(b, a\%b) \tag{10}$$

Implementation (C#)

```
/// <summary>
/// Implementation of Euclids algorithm
/// </summary>
/// <param name="a"></param>
/// <param name="b"></param>
/// <returns></returns>
public static int HighestCommonFactor(int a, int b)
{
  if (a < b)
  {
    return HighestCommonFactor(b, a);
  }
  else
  {
    int remainder = a % b;

    if (remainder == 0)
    {
      return b;
    }
    else
    {
      return HighestCommonFactor(b, remainder);
    }
  }
}
```

Floor/Ceiling Functions

Definitions

Floor – The greatest integer less than x

$$floor : \mathcal{R} \longrightarrow \mathbb{Z}$$

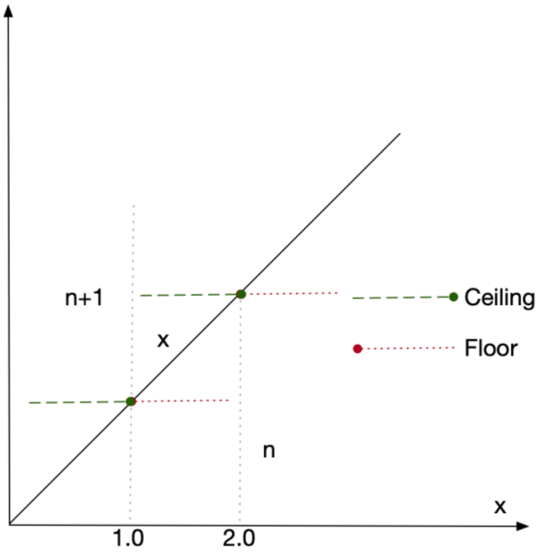
$$floor(x) = \lfloor x \rfloor = max\{a \in \mathbb{Z} | \; a \leq x\}$$

Ceiling – The smallest integer less than x

$$ceiling : \mathcal{R} \longrightarrow \mathbb{Z}$$

$$ceiling(x) = \lceil x \rceil = max\{a \in \mathbb{Z} | \; a \leq x\}$$

FIGURE 2FLOOR/CEILING



LISTING 1 EXAMPLES

- $\lfloor 1.0 \rfloor = \lceil 1.0 \rceil = 1$
- $\lfloor 1.0000001 \rfloor = 1$
- $\lceil 1.0000001 \rceil = 2$
- $\lfloor 1.9999999 \rfloor = 1$
- $\lceil 1.9999999 \rceil = 2$

Fractional Part

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

$\{x\} = x - \lfloor x \rfloor$

We can use this notation to calculate possible values of $\lfloor x + y \rfloor$

Property Summary

1. $\lfloor \mathrm{x} \rfloor = \lceil \mathrm{x} \rceil = \mathrm{x} \leftrightarrow \mathrm{x} \in \mathbb{Z}$

2. $x - 1 < \lfloor x \rfloor \leq \lceil x \rceil < x + 1, \qquad x \in \mathbb{R}$

3. $\lfloor -x \rfloor = -\lceil x \rceil, \qquad x \in \mathbb{R}$

4. $-\lfloor x \rfloor = \lceil -x \rceil, \qquad x \in \mathbb{R}$

5. $\lceil x \rceil - \lfloor x \rfloor = 0 \leftrightarrow x \in \mathbb{Z}$

6. $\lceil x \rceil - \lfloor x \rfloor = 1 \leftrightarrow x \notin \mathbb{Z}$

7. $\lfloor x \rfloor = n \leftrightarrow \mathrm{n} \leq x < n + 1, x \notin \mathbb{R}, \mathrm{n} \notin \mathbb{Z}$

8. $\lceil x \rceil = n \leftrightarrow \mathrm{n} - 1 < x \leq, x \notin \mathbb{R}, \mathrm{n} \notin \mathbb{Z}$

9. $\lfloor x \rfloor = n \leftrightarrow x - 1 < n \leq, \notin \mathbb{R}, \mathrm{n} \notin \mathbb{Z}$

10. $\lceil x \rceil = n \leftrightarrow \mathrm{x} \leq n < x + 1, x \notin \mathbb{R}, \mathrm{n} \notin \mathbb{Z}$

11. $\left\lceil \frac{\mathrm{x}}{2} \right\rceil + \left\lfloor \frac{\mathrm{x}}{2} \right\rfloor = \mathrm{x}$

12. $\lfloor x + n \rfloor = \lfloor x \rfloor + n$, if n is an integer and x a real

13. $\lceil x + n \rceil = \lceil x \rceil + n$, if n is an integer and x a real

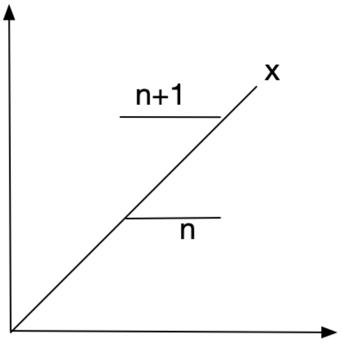
14. $x < n \rightarrow \lfloor x \rfloor < n$, if n is an integer and x a real

15. $n < x \rightarrow n < \lceil x \rceil$ if n is an integer and x a real

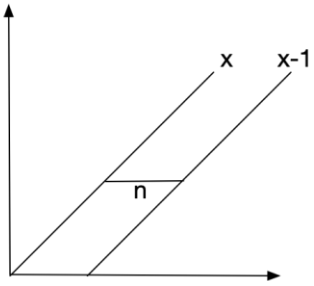
16. $\lceil f(\lceil x \rceil) \rceil = \lceil f(x) \rceil$ if f is continuous, monotonically increasing with the property that if $f(x) \in \mathbb{Z}$ then $x \in \mathbb{Z}$

Property Detail

PROPERTY 7 $\lfloor \boldsymbol{x} \rfloor = \boldsymbol{n} \leftrightarrow \mathbf{n} \leq \boldsymbol{x} < \boldsymbol{n} + \mathbf{1}, \boldsymbol{x} \notin \mathbb{R}, \mathbf{n} \notin \mathbb{Z}$



PROPERTY 8 $\lceil \boldsymbol{x} \rceil = \boldsymbol{n} \leftrightarrow \mathbf{n} - \mathbf{1} < \boldsymbol{x} \leq, \boldsymbol{x} \notin \mathbb{R}, \mathbf{n} \notin \mathbb{Z}$



Property 16

If we define function

$$f\colon \mathbb{R}' \longrightarrow \mathbb{R}\,|\,\mathbb{R}' \subseteq\ \mathbb{R}\ is\ the\ domain\ of\ f$$

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

Property P: if $f(x) \in \mathbb{Z}$ then $x \in \mathbb{Z}$

Then for all $x \in \mathbb{R}'$ for which the property P holds

$\lceil f(\lceil x \rceil) \rceil = \lceil f(x) \rceil$

Proof

In the simple case where $x = \lceil x \rceil$ we have nothing to do. We hence focus on the case where $x \neq \lceil x \rceil$.

$x \neq \lceil x \rceil \rightarrow x \leq \lceil x \rceil$ *From the definition of the ceiling function*

$x \leq \lceil x \rceil \rightarrow f(x) \leq f(\lceil x \rceil)$ *Because f is monotonically increasing*

$f(x) \leq f(\lceil x \rceil) \rightarrow \lceil f(x) \rceil \leq \lceil f(\lceil x \rceil) \rceil$ *Because ceiling is non decreasing*

Assume

$\lceil f(x) \rceil < \lceil f(\lceil x \rceil) \rceil$

$\lceil f(x) \rceil < \lceil f(\lceil x \rceil) \rceil \rightarrow \lceil f(\lceil x \rceil) \rceil - \lceil f(x) \rceil \geq 1$ *Because ceiling only deals in integers*

This means the monotonically increasing function f must increase above $\lceil f(x) \rceil$ in order to make it possible for $\lceil f(\lceil x \rceil) \rceil - \lceil f(x) \rceil \geq$ This means that the following two things must be true

$\forall y |\ x \leq y < \lceil x \rceil$

$f(y) = \lceil \mathrm{f}(x) \rceil$

The special property P means that y must be an integer as $\lceil \mathrm{f}(x) \rceil$ is by definition an integer. But there cannot be an integer between x and $\lceil x \rceil$ so we have a contradiction and hence it is not possible for

$\lceil f(x) \rceil < \lceil f(\lceil x \rceil) \rceil$ and hence $\lceil f(x) \rceil = \lceil f(\lceil x \rceil) \rceil$