Design & Analysis of Algorithm

Introduction to Algorithms by Cormen
Number Theoretic Algorithms

Chapter 31

Divisors

Let *a*, *b* and *c* be integers such that

$$a = b \cdot c\,.$$

Then *b* and *c* are said to ***divide*** (or are ***factors***) of *a*, while *a* is said to be a ***multiple*** of *b* (as well as of *c*). The pipe symbol "|" denotes "divides" so the situation is summarized by:

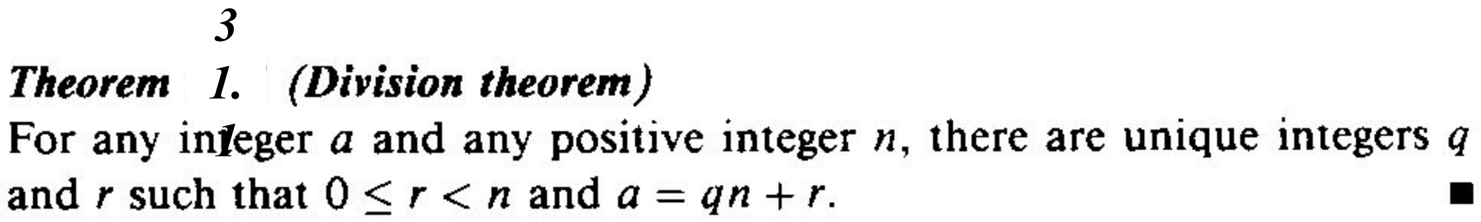
$$b \mid a \;\wedge\; c \mid a\,.$$

Divisors Examples

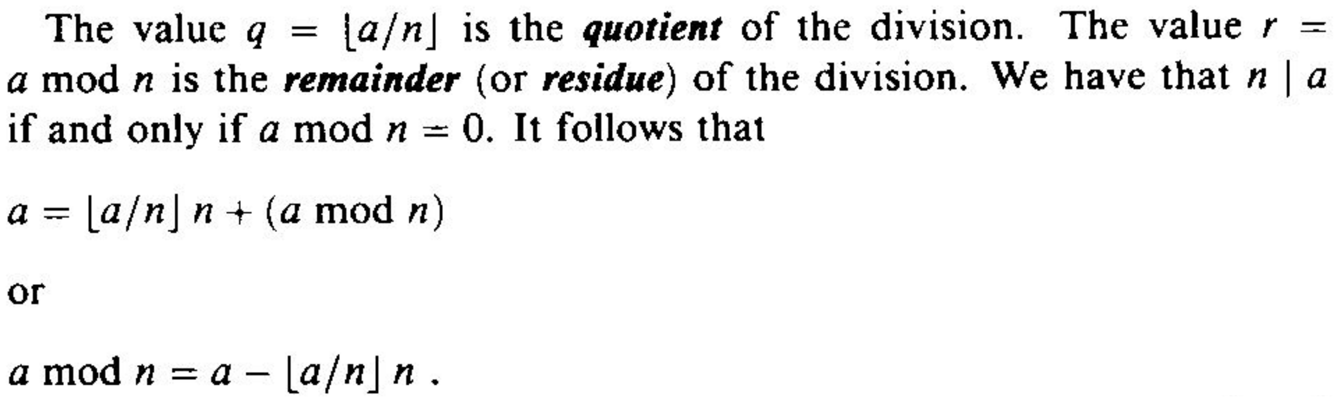
Q: Which of the following is true?

1. $77 \mid 7$
2. $7 \mid 77$
3. $24 \mid 24$
4. $0 \mid 24$
5. $24 \mid 0$

Division & Remainders



***Theorem 31.1** (Division theorem)*
For any integer a and any positive integer n, there are unique integers q and r such that $0 \le r < n$ and $a = qn + r$. ■



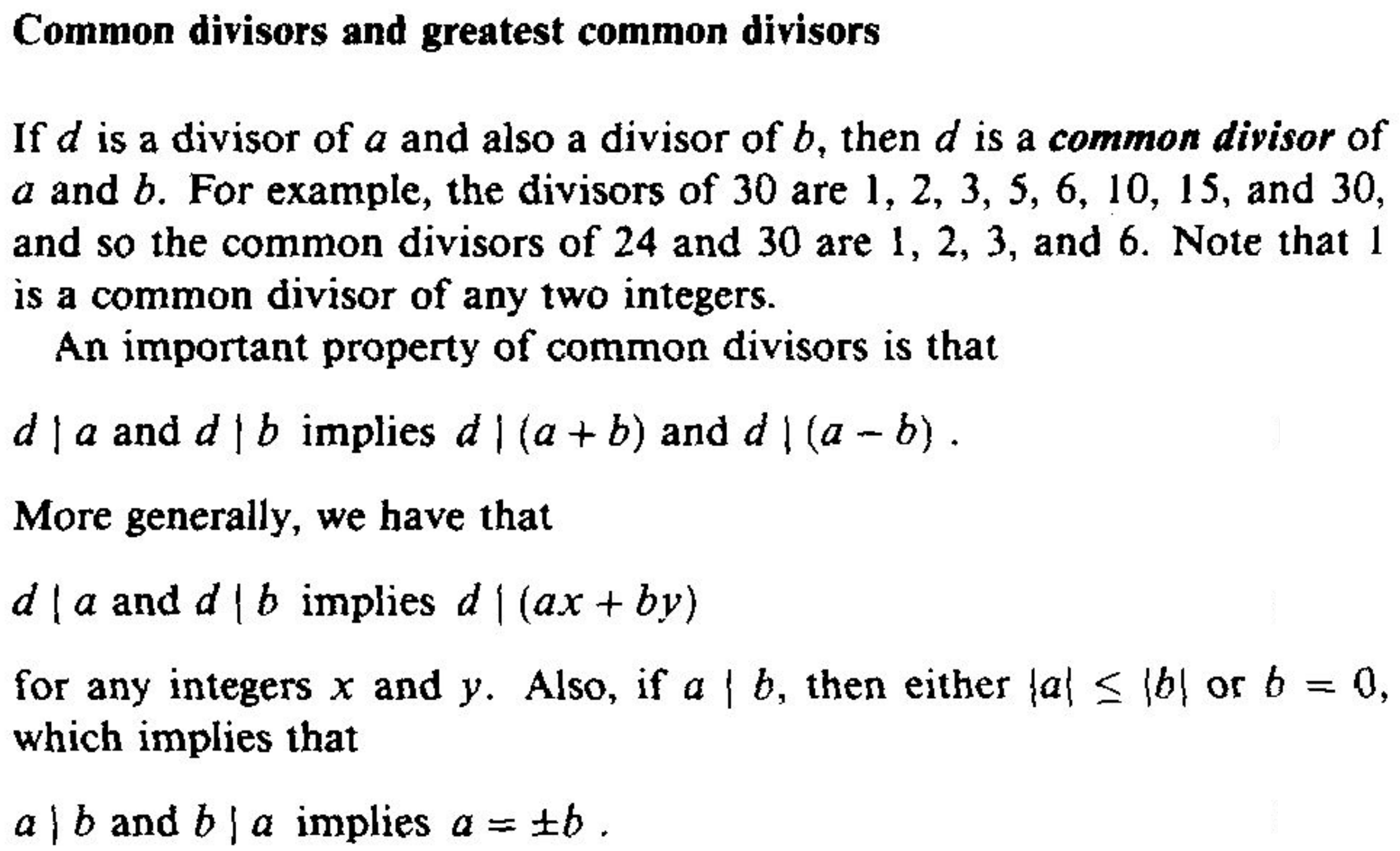
The value $q = \lfloor a/n \rfloor$ is the ***quotient*** of the division. The value $r = a \bmod n$ is the ***remainder*** (or ***residue***) of the division. We have that $n \mid a$ if and only if $a \bmod n = 0$. It follows that

$$a = \lfloor a/n \rfloor n + (a \bmod n)$$

or

$$a \bmod n = a - \lfloor a/n \rfloor n \ .$$

Common Divisors



Common divisors and greatest common divisors

If d is a divisor of a and also a divisor of b, then d is a ***common divisor*** of a and b. For example, the divisors of 30 are 1, 2, 3, 5, 6, 10, 15, and 30, and so the common divisors of 24 and 30 are 1, 2, 3, and 6. Note that 1 is a common divisor of any two integers.

An important property of common divisors is that

$$d \mid a \text{ and } d \mid b \text{ implies } d \mid (a+b) \text{ and } d \mid (a-b)\ .$$

More generally, we have that

$$d \mid a \text{ and } d \mid b \text{ implies } d \mid (ax+by)$$

for any integers x and y. Also, if $a \mid b$, then either $|a| \le |b|$ or $b = 0$, which implies that

$$a \mid b \text{ and } b \mid a \text{ implies } a = \pm b\ .$$

Greatest Common Divisor
Relatively Prime

- Let a,b be integers, not both zero. The ***greatest common divisor*** of a and b (or gcd(a,b)) is the biggest number d which divides both a and b.
- a and b are said to be ***relatively prime*** if gcd(a,b) = 1, so no prime common divisors.

Greatest Common Divisor Relatively Prime

Q: Find the following gcd's:

1. gcd(11,77)
2. gcd(33,77)
3. gcd(24,36)
4. gcd(24,25)

Greatest Common Divisor Relatively Prime

A:

1. gcd(11,77) = 11
2. gcd(33,77) = 11
3. gcd(24,36) = 12
4. gcd(24,25) = 1. Therefore 24 and 25 are relatively prime.

NOTE: A prime number are relatively prime to all other numbers which it doesn't divide.

Greatest Common Divisor Relatively Prime

EG: More realistic. Find gcd(98,420).

Find prime decomposition of each number and find all the common factors:

$98 = 2\cdot 49 = 2\cdot 7\cdot 7$

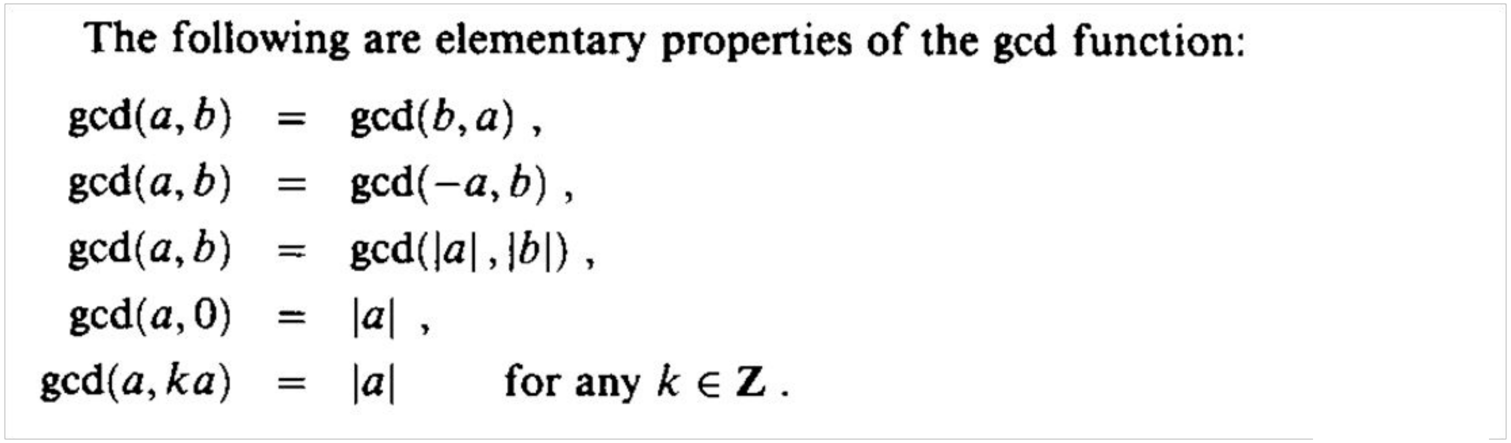
$420 = 2\cdot 210 = 2\cdot 2\cdot 105 = 2\cdot 2\cdot 3\cdot 35$

$\quad = 2\cdot 2\cdot 3\cdot 5\cdot 7$

Underline common factors: $\underline{2\cdot 7}\cdot 7$, $2\cdot \underline{2}\cdot 3\cdot 5\cdot \underline{7}$

Therefore, gcd(98,420) = 14

Greatest Common Divisor



The following are elementary properties of the gcd function:

$$\begin{aligned}
\gcd(a,b) &= \gcd(b,a)\ , \\
\gcd(a,b) &= \gcd(-a,b)\ , \\
\gcd(a,b) &= \gcd(|a|,|b|)\ , \\
\gcd(a,0) &= |a|\ , \\
\gcd(a,ka) &= |a| \qquad \text{for any } k \in \mathbf{Z}\ .
\end{aligned}$$

The GCD and Linear Combinations

- **Theorem 31.2:**
  If a and b are integers not both 0, then gcd(a, b) is the smallest positive element of the set $\{ax + by : \text{x, y are integers}\}$ of linear combinations of a and b.

  Proof: see text p. 853.

The GCD and Linear Combinations (2)

- Corollaries:
  - For any integers a and b, if $d|a$ and $d|b$ then $d|\gcd(a, b)$.
  - For all integers a and b and any nonnegative integer n, $\gcd(an, bn) = n \gcd(a, b)$.
  - For all positive integers n, a, and b, if $n|ab$ and $\gcd(a, n) = 1$, then $n|b$.

Relatively Prime Integers

- Two integers a and b are *relatively prime* if and only if their only common divisor is 1
  (i.e., $\gcd(a, b) = 1$).
- **Theorem 31.6:**
  For any integers a, b, and p, if both $\gcd(a, p) = 1$ and $\gcd(b, p) = 1$, then $\gcd(ab, p) = 1$.
  Proof: p. 854 in text.

Divisibility by Primes

- **Theorem 31.7:**
  For all primes p and all integers a, b, if $p|ab$, then $p|a$ or $p|b$ (or both).

  Proof: p. 854 in text

The GCD Recursion Theorem

- **Theorem 31.9 (GCD recursion theorem):**
  For any nonnegative integer a and any positive integer b,

  $$\gcd(a, b) = \gcd(b, a \bmod b).$$

  Proof: p. 857 of the text.

Euclid's Algorithm – Finding GCD

- Based on the following theorem
  - gcd(a, b) = gcd(b, a mod b)
  - Proof
    - If d = gcd(a, b), then d|a and d|b
    - For any positive integer b, a = kb + r ≡ r mod b, a mod b = r
    - a mod b = a – kb (for some integer k)
      - because d|b, d|kb
      - because d|a, d|(a mod b)

    ∴ d is a common divisor of b and (a mod b)
    - Conversely, if d is a common divisor of b and (a mod b), then d|kb and d|[kb+(a mod b)]
    - d|[kb+(a mod b)] = d|a

    ∴ Set of common divisors of a and b is equal to the set of common divisors of b and (a mod b)
    - ex) gcd(18,12) = gcd(12,6) = gcd(6,0) = 6

      gcd(11,10) = gcd(10,1) = gcd(1,0) = 1

Euclid's GCD Algorithm

```
Euclid($a$, $b$):
  if ($b$ == 0)
    return $a$
  else
    return Euclid($b$, $a$ mod $b$)
```

Euclid's Algorithm – Finding GCD

- Recursive algorithm

```
Function Euclid (a, b)        /* assume a ≥ b ≥ 0 */
    if b = 0 then return a
        else return Euclid(b, a mod b)
```

- Iterative algorithm

```
Euclid(d, f)                  /* assume d > f > 0 */
1.  X ← d; Y ← f
2.  if Y=0  return X = gcd(d, f)
3.  R = X mod Y
4.  X ← Y
5.  Y ← R
6.  goto 2
```

Euclid's Algorithm Example

Euclid(1155, 546) =

Euclid(546, 63) =

Euclid(63, 42) =

Euclid(42, 21) =

Euclid(21, 0)

The gcd of 1155 and 546 is 21.

Properties of Euclid's Algorithm

- Since the second argument is monotonically decreasing, and the gcd is positive, the algorithm terminates.

- Theorem 31.9 implies that the algorithm computes the gcd correctly.

Extended Euclidean Algorithm

- You are given two integer number a and b. Find integer coefficients x and y such that

$$d=\gcd(a, b) = ax+by$$

- The extended Euclidean algorithm works the same as the regular Euclidean algorithm except that we keep track of more details –namely the quotient $q = a/b$ in addition to the remainder $r = a$ **mod b**. This allows us to backtrack and write the gcd(*a*,*b*) as a linear combination of *a* and *b*.

Extended Euclid's Algorithm

- ExtEuclid(a, b):

```
**if** $b == 0$
    **return** $(a, 1, 0)$
$(d', x', y') \leftarrow$ ExtEuclid($b$, $a \bmod b$)
$(d, x, y) \leftarrow (d', y', x' - \text{floor}(a/b) \cdot y')$
**return** $(d, x, y)$
```

- $d = \gcd(a, b) = ax + by$

Extended Euclid's Algorithm

Suppose a and b are given. Find x and y such that,

$$ax+by=\gcd(a,b)$$

Let, $d=\gcd(a,b)$

Then, $ax+by=d$ [by Thm 31.2]

Again, $\gcd(a,b)=\gcd(b, \mathbf{a \bmod b})$

So, $ax+by = d = bx'+(\mathbf{a \bmod b})y'$

We know that, $a = b.\lfloor a/b \rfloor + a \bmod b$

So, $\mathbf{a \bmod b} = a - b.\lfloor a/b \rfloor$

Thus, $ax+by = bx'+(a - b.\lfloor a/b \rfloor).y'$

$$= ay'+b(x' - \lfloor a/b \rfloor.y')$$

Finally, $x = y'$

$y = x' - \lfloor a/b \rfloor.y'$

Finding Prime Factors

- How can we calculate the primes between [1, N] ?
- Can we determine if a given number K is prime or not
  - using [2-K-1] loop
  - using (K/2) loop
  - using sqrt(K) loop
    - K = c*d, [c=d=sqrt(K)], [c>sqrt(K), d<sqrt(K)]

Sieve of Eratosthenes

- An efficient technique to find the primes within [1,N]
- idea is to clip out multiples of primes

Using Sieve to detect primes

- Will calculate the primes between [1,30]
- Define an array status[31]
  - if status[i] == 0, i is prime
  - if status[i] == 1, i is not prime
  - initially assume every i is prime

Using Sieve to detect primes

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30

Using Sieve to detect primes

- 1 is not prime, status[1] = 1
- 2 is prime
- multiples of 2 is not prime, set status for every 2*i as not prime

Using Sieve to detect primes

1	🟥 2	🟥 3	4	🟥 5
6	🟥 7	8	🟥 9	10
🟥 11	12	🟥 13	14	🟥 15
16	🟥 17	18	🟥 19	20
🟥 21	22	🟥 23	24	🟥 25
26	🟥 27	28	🟥 29	30

For (i=2, i<=N;i=i+2) status[i] = 1

Using Sieve to detect primes

- Now iterate over the odd numbers
- If it is prime, untrack its multiples from being prime

For a number N, it is enough to look within sqrt(N), because it will at least have one prime number within that range which is its factor.

```
For (i = 3;  i<=sqrt(N); i = i+2) {
     if (status[i] == 0) {
          For(j = 2 * i ; j <=N; j = j + i) {
               status[j] = 1
          }
     }
}
```

Using Sieve to detect primes

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30

For i = 3

Using Sieve to detect primes

1	🟥 2	🟥 3	4	🟥 5
6	🟥 7	8	9	10
🟥 11	12	🟥 13	14	15
16	🟥 17	18	🟥 19	20
21	22	🟥 23	24	25
26	27	28	🟥 29	30

For i = 5

Using Sieve to detect primes

```
For (i = 3;  i<=sqrt(N); i = i+2) {
     if (status[i] == 0) {
          For(j = i * i ; j <=N; j = j + 2 * i) {
               status[j] = 1
          }
     }
}
```

(i+1) is even which is already been detected from 2

Optimization: because smaller multiples already been sieved/cut

Utilities

- After tracking the primes, it is very easy to factorize each number
- let such array/vector be primes

```
For (i=0; i<vector.size() && primes[i] <= sqrt(N); i++) {
        while (N%primes[i] == 0) {
                N = N/primes[i] // detecting powers + primes
        }
}
if N is not 1, this remaining factor of N is a prime.
```