3.1, 3.2, 3.3 Algorithms

Computer Science is the study of Algorithms (Knuth)

Algorithm ≡ A finite set of precise instruction for performing computation or for solving a problem

Computational complexity – time and space

Time-complexity:

by counting the number of operation

Frequency Count

Ex.
```
for i:= 1 to n do
  for j:= 1 to i do
    x:= x + 1    (*)
  endfor
endfor
```

i	j
1	1
2	1 2
3	1 2 3
:	:
n	1 2 : n

$$1 + 2 + 3 + \ldots + n = \frac{n(n+1)}{2}$$

Challenge: What is the value of x when the following program end?

```
x:= 0
for i:= 1 to n do
  for j:= 1 to i do
    for k:= 1 to j do
      x:= x + 1
    endfor
  endfor
endfor
```

Hint: $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

Sorting and Searching

The Art of Computer Programming, Vol 3 (722 pages)

Donald Knuth, 1973

- Sorting by insertion
  - Straight insertion
  - Two-way insertion
  - Diminishing increment sort (Shell's)
  - List insertion
  - Multiple list insertion
- Sorting by exchanging
  - Bubble sort
  - Merge-exchange sort (Batcher's)
  - Partition-exchange sort (Quicksort)
  - Radix-exchange sort
- Sorting by selection
  - Straight selection
  - Tree selection
  - Heapsort
- Sorting by merging
  - Straight two-way merge sort
  - Natural two-way merge sort
  - List merge sort
- Sorting by distribution
  - Radix list sort
  - Hooking-up of queue

Searching

1. Linear Search (Sequential search)

```
Procedure linear_search (x, a1,a2, ..., an, Loc)
    i:= 1
    while (i ≤ n and x ≠ ai ) do
        i:= i + 1
    endwhile
    if i ≤ n  then  Loc:= i
              else  Loc:= 0
    endif
```

Time-complexity:

Number of comparisons = 2n + 1

Note. if x is not in the list, 2n + 2

2. Binary Search (when a list is already sorted)

Idea. Compare with middle one in the list. Decide which sublist that x belong. Repeat the process until |List| become 1.

```
Procedure binary_search (x, a1,a2, ..., an, Loc)
    i:= 1   // left end point //
    j:= n   // right end point //
    while  i < j  do
        m:= ⌊(i + j) / 2⌋

        if  x > am then  i:= m + 1
                  else  j:= m
        endif
    endwhile
    if  x = ai  then Loc:= i
                else  Loc:= 0
```

Ex 3. $x = 19,\ n = 16$

$L = [\ 1\ 2\ 3\ 5\ 6\ 7\ 8\ \textcircled{10}\ 12\ 13\ 15\ 16\ 18\ 19\ 20\ 22\ ]$

$10 < 19 \rightarrow$ right

$[\ 12\ 13\ 15\ \textcircled{16}\ 18\ 19\ 20\ 22]$

$16 < 19 \rightarrow$ right

$[\ 18\ \textcircled{19}\ 20\ 22\ ]$

found!

Time-complexity

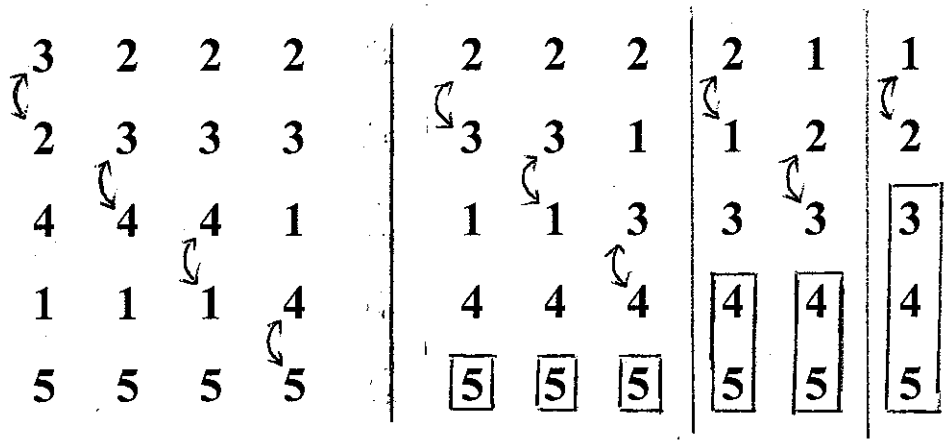
$n = 2^k$ $(k = \log n)$

1^{st} stage	$(\lvert L\rvert = 2^k)$:	2
2^{nd} stage	$(\lvert L\rvert = 2^{k-1})$:	2
:	:	:
k stage	$(\lvert L\rvert = 2^1)$	2

$2k + 2 = 2 \log n + 2$

Bubble Sort

L = [3 2 4 1 5]



Procedure bubble_sort(a_1, a_2, …, a_n)

```
for i:= 1 to n-1 do
    for j:= 1 to n-i do
        if a_j > a_{j+1} then a_j ↔ a_{j+1} endif
    endfor
endfor
```

Time-complexity

$$(n-1) + (n-2) + \ldots + 2 + 1 = \frac{(n-1)\cdot n}{2}$$

T(n) =

Insertion Sort

L = [3 2 4 1 5]

	Forward Scan	Backward Scan
i=1	3 | 2 4 1 5	3 | 2 4 1 5
i=2	3 ↔ 2 | 4 1 5	3 ↔ 2 | 4 1 5
	2 3 | 4 1 5	2 3 | 4 1 5
i=3	2 ↔ 3 ↔ 4 | 1 5	2 3 ↔ 4 | 1 5
	2 3 ↔ 4 | 1 5	
i=4	2 ↔ 3 4 ↔ 1 | 5	2 3 4 ↔ 1 | 5
	1 2 3 4 | 5	2 3 ↔ 1 4 | 5
		2 ↔ 1 3 4 | 5
		1 2 3 4 | 5
i=5	1 ↔ 2 3 4 ↔ 5	1 2 3 4 ↔ 5
	1 2 ↔ 3 4 ↔ 5	
	1 2 3 ↔ 4 ↔ 5	
	1 2 3 4 ↔ 5	

Time-complexity: $1 + 2 + \ldots (n-1) = \frac{n(n-1)}{2}$ $O(n^2)$

Growth of Functions (Asymptotic rate of Growth)

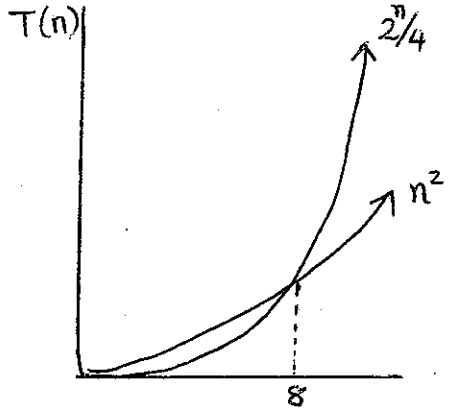
Motivation

Ex. n^2 vs. $2^n/4$

$n < 8 \quad n^2 > 2^n/4$

$n = 8 \quad n^2 = 2^n/4$

$n > 8 \quad n < 2^n/4$



O: upper bound

Def: $f(n) = O(g(n))$ iff there exists constants C and k such that $|f(n)| \leq C\,|g(n)|$ for $n \geq k$

note. We need to find only one pair of constants C and k.

Ω: lower bound

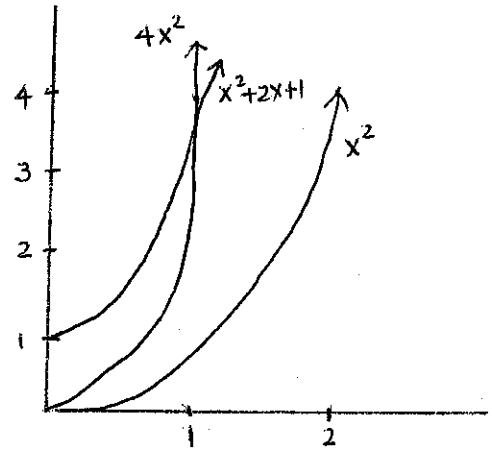
Def: $f(n) = \Omega(g(n))$ iff there exists constants C and k such that $|f(n)| \geq C\,|g(n)|$ for $n \geq k$

Θ: exact bound

Def: $f(n) = \Theta(g(n))$ iff there exists positive constants C_1, C_2, k such that $C_1 g(n) \leq f(n) \leq C_2 g(n)$ for $n \geq k$

Note O: asymptotic

Ex 1. $f(x) = x^2 + 2x + 1$ **is** $O(x^2)$



Note. when $x > 1$, $x^2 > x > 1$

$$0 \leq x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$$

Choose $C = 4, k = 1$

Note. f(x) is $O(x^3)$, $O(x^3)$, … **We choose the tightest one.**

Ex. $f(n) = 5n^3 + 2n^2 + 22n + 6$
$f(n) = O(n^3)$

Proof:

Let C = 6 (why?). We want to find the smallest n such that

$6n^3 > 5n^3 + 2n^2 + 22n + 6$

$n^3 > (2n + 22n + 6)$

$n = 1 \quad 1 < 30$
$n = 2 \quad 8 < 126$
$\vdots \quad \vdots$
$n = 5 \quad 125 < 156$

$n = 6 \quad 216 > 210$
$n = 7 \quad 343 > 258$
$\vdots \quad \vdots > \vdots$

$C = 6, \; k = 6$ q.e.d

Note.

$O(n^5)$	$\Omega(n)$
$O(n^4)$	$\Omega(n^2)$
$O(n^3)$	$\Omega(n^3)$
---	---
$\times \; O(n^2)$	$\times \; \Omega(n^4)$

$5n^3 \leq (5n^3 + 2n^2 + 22n + 6) \leq 6n^3$ for $n \geq k$

$\therefore \; \Theta(n^3)$

Ex 2. Show that $7x^2$ is $O(x^3)$

When $x > 7$, then $7x^2 < x^3$
Choose $C = 1, k = 7$

Thm 1. Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + + \ldots + a_1 x + a_0$
Then $f(n) = O(x^n)$

Ex 5. $1 + 2 + \ldots + n$

$$1 + 2 + \ldots + n \leq n + n + \ldots n = n^2$$

$$\therefore\ 1 + 2 + \ldots n = O(n^2),\quad C = 1, k = 1$$

Ex. 6 $n! = 1 \cdot 2 \cdot 3 \cdots n$

$$n! = 1 \cdot 2 \cdot 3 \cdots n \leq n \cdot n \cdot n \ldots n = n^n$$
$$\therefore\ n! = O(n^n),\quad C = 1, k = 1$$

Note. $\log n! \leq \log n^n = n \log n$

Ex 8. $f(x) = 3n \log(n!) + (n^2 + 3) \log n$

$3n \log(n!)$: $O(n \log n)$ → $O(n^2 \log n)$
$(n^2 + 3) \log n$: $O(n^2 \log n)$

Ans: $O(n^2 \log n)$

Ex 9. $f(x) = (x + 1) \log (x^2 + 1) + 3x^2$

$(x + 1) \log (x^2 + 1)$: $x \log x$
$3x^2$: x^2

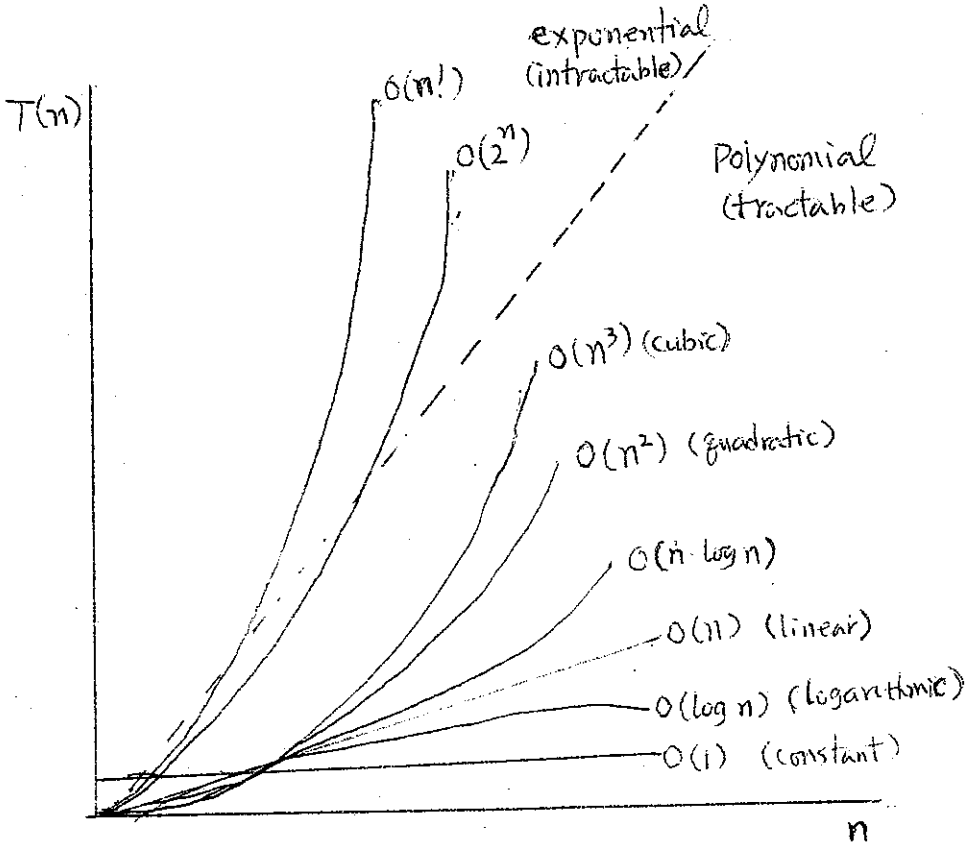
$O(x^2)$

Ex. 10 $f(x) = 8x^3 + 5x^2 + 7$

$f(x)$ is $\Omega(g(x))$ where $g(x) = x^3$

$f(x) = 8x^3 + 5x^2 + 7 > 8x^3$

$C = 8, k = 1$



3.4 Integers and Division

Mathematics – set theory, topology, *number theory*, …

Number theory – division, gcd, prime number

Integer division

Def. $a \mid b$ **(a divides b)** // $\exists\, c\ (ac = b)$ //

Thm 1.

1. **if** $a \mid b$ **and** $a \mid c$, **then** $a \mid (b + c)$
2. **if** $a \mid b$, **then** $a \mid bc$ **for all integer c**
3. **if** $a \mid b$ **and** $b \mid c$, **then** $a \mid c$

Corollory.

$a \mid b$ **and** $a \mid c \rightarrow a \mid mb + nc$

Division Algorithm

$a = d\,\underline{q} + \underline{r}$ // unique q and r //

$q = a \text{ div } d$ // quotient //

$r = a \bmod d$ // remainder //

Ex. 4 (-11 div 3)

$-11 = 3(-4) + 1$ (O)

$-11 = 3(-3) + (-2)$ (X) // remainder be 0,1,2 //

Modular Arithmetic

Def. $a = b \pmod m$ // a is congruent to b modulo m //

Thm 3. $a \equiv b \pmod m$ iff $a \bmod m = b \bmod m$

Thm 4. $a = b \pmod m$ iff $a = b + km$

Proof.

(i) $a = b \pmod m$,
then $m \mid (a-b)$
$a-b = km$
$a = b+km$

(ii) $a = b+km$
$km = a-b$
$\therefore k$ divides $(a-b)$
$a = b \pmod m$

Thm 5. If $a \equiv b \pmod m$ and $c \equiv d \pmod m$, then
$a + c \equiv b + d \pmod m$ and $ac \equiv bd \pmod m$

Ex 6 $7 \equiv 2 \pmod 5$ $\quad 11 \equiv 1 \pmod 5$
$7 + 11 = 2 + 1 \pmod 5$ and $7 \cdot 11 = 2 \cdot 1 \pmod 5$
// $18 = 3 \pmod 5$ and $77 = 2 \pmod 5$ //

Applications of Congruence

(1) Hash function

Ex.

$h(k) = k \bmod m$ // k: SSN //
// m: memory location , eg 111 /

$h(064212848) = 064212848 \bmod 111 = 14$
$h(037149212) = 037149212 \bmod 111 = 64$
: : : = :
$h(107405723) = 107405723 \bmod 111 = 14$ → *collision*

(2) Random number generator

- generates random numbers between 0 and 1
- computer simulation
- pseudo random generator

Linear congruential method

$$x_{n+1} = (a\, x_n + c) \bmod m$$

Ex.8 $m = 9,\ a = 7,\ c = 4,\ x_0 = 3$

$X_1 = (7 \cdot 3 + 4) \bmod 9 = 7$ (***)
$X_2 = (7 \cdot 7 + 4) \bmod 9 = 8$
$X_3 = (7 \cdot 8 + 4) \bmod 9 = 6$
$X_4 = (7 \cdot 6 + 4) \bmod 9 = 1$
$X_5 = (7 \cdot 1 + 4) \bmod 9 = 2$
$X_6 = (7 \cdot 2 + 4) \bmod 9 = 0$
$X_7 = (7 \cdot 0 + 4) \bmod 9 = 4$
$X_8 = (7 \cdot 4 + 4) \bmod 9 = 5$
$X_9 = (7 \cdot 5 + 4) \bmod 9 = 3$
$X_{10} = (7 \cdot 3 + 4) \bmod 9 = 7$ (***) cycle

Note. $m = 2^{31} - 1,\quad a = 16{,}807,\quad c = 0$

Note. Testing random number generators

Empirical test – χ^2 test (chi square test)
Theoretical test
- spectral test
- lattice test

:

Testing Random Number Generators

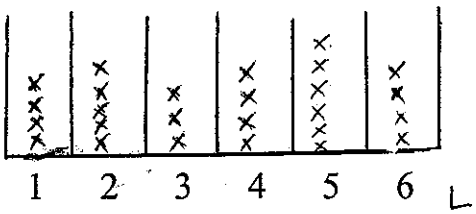
- uniformity and independence

Empirical Tests

(1) Frequency Test (Equidistance Test)

- uniformity

Ex. Die



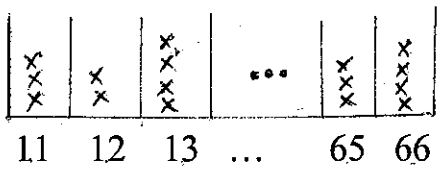
$\frac{n_i}{N} \doteq \frac{1}{L}$

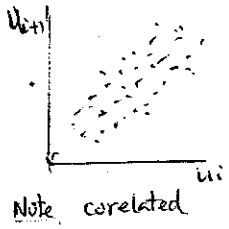
K-S test

χ^2 test

(2) Serial Test

- independence and uniformity using two consecutive digits





(2)' Serial Relationship Test

- $\frac{\text{random numbers falling into two consecutive intervals}}{N} = \frac{1}{L}$

Ex 3 6 1 2 5 4 ...

(3) Gap Test
- distance between 0's

Ex. gap length 0: 00
gap length 1: 070
gap length 2: 0850
gap length 3: 03140
: :

(4) Poker Test
- sequence of 5 numbers based on poker hands

(5) Coupon's Collector Test
(6) Permutation Test

(7) Run Test
- count the number of sequences of length 1,2,3,4,5, or ≥6, where the values within those sequences are monotonically increasing.

Ex.
(0.86), (0.11, 0.23), (0.03, 0.13), (0.06, 0.55, 0.64, 0.87), (0.71) ..

r1 =2, r2 = 2, r3 = 0, r4 = 1, r5 = 0, r6 = 0

- approximate chi-square distribution with 6 df

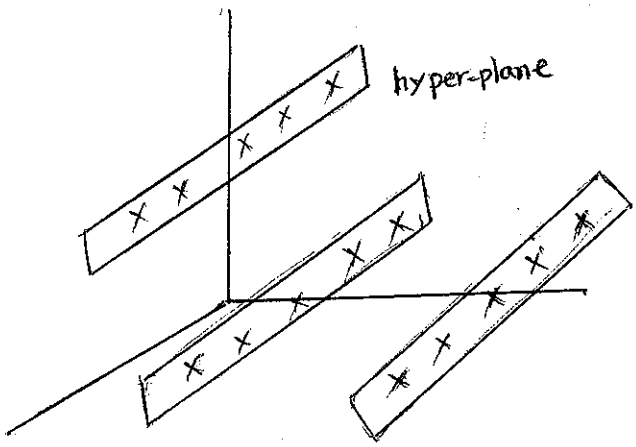
$$\underline{R} = 1/n \sum_{i=1}^{6} \sum_{j=1}^{6} a_{ij} (r_i - n \cdot b_i)(r_j - n \cdot b_j)$$

(8) Maximum-of-t Test
(9) Collision Test
:

Theoretical Test

(1) Spectral Test

(2) Lattice Test – check the gap between hyper-planes
(Dimension 2 through dimension 10)



Note. There is no "the best" generator in absolute sense

(3) Cryptology

Caesar's encryption method

$f(p) = (p + 3) \bmod 26$ // shift cipher //

0 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
a b c d e f g h i j k l m n o p q r s t u v w x y z

x y z a b c d e f g h i j k l m n o p q r s t u v w
0 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

Ex. "attack at dawn"

Encription: $f(p) = (p + 3) \bmod 26$

0 19 19 0 2 10 0 19 3 0 22 13
3 22 22 3 5 13 3 22 6 3 25 16

"d w w d f n d w g d z q"

Decription: $f(p) = (p - 3) \bmod 26$

3.5 Primes and GCD

Primes

2, 3, 5, 7, 11,13, 17, 19, ...

Thm 1. for $n > 1$, n can be written uniquely as a prime or as the product of 2 or more primes.

Ex. $90 = 2\,3\,3\,5 = 2\,3^2\,5$

$1024 = 2\,2 \ldots 2 = 2^{10}$

Thm 2. If n is a composite integer, then n has a prime divisor less than or equal to $\sqrt{n}$.

Proof.

$n = ab$

$a \leq \sqrt{n}$ or $b \leq \sqrt{n}$ (otherwise, $ab > \sqrt{n}\cdot\sqrt{n} = n$)

$\therefore$ n has a divisor not exceeding $\sqrt{n}$.

the divisor is either prime, or has a prime divisor.

n has a prime divisor $< \sqrt{n}$

Ex 3. Show that 101 is prime.

$\sqrt{101} = \lfloor 10.049.. \rfloor = 10$

Divisors to check: 2, 3, 5, 7 → prime

Thm 3. There are infinitely many primes.

Mersenne prime: $2^p - 1$

GCD and LCM

Ex. gcd(18,30) = 6
gcd(17,22) = 1 // relatively prime //

Ex. 12, 7, 5 // pairwise relatively prime //

Ex. $120 = 2^3 \cdot 3 \cdot 5$
$500 = 2^2 \cdot 5^3$

$\gcd(120,500) = 2^2 \cdot 5 = 20$
$\text{lcm}(120,500) = 2^3 \cdot 3 \cdot 5^3 = 3000$

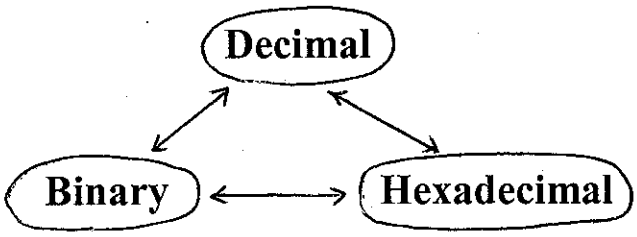
Thm 5. $ab = \gcd(a,b) \cdot \text{lcm}(a,b)$

3.6. Integers and Algorithms

Number systems

Decimal	0,1,2,3,4,5,6,7,8,9
Binary	0,1
Octal	0,1 2,3 4,5,6,7
Hexadecimal	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Number conversion



Ex. B → D

$$(1\,0\,0\,1\,1\,0\,1\,1)_2 = 1\cdot2^7 + 1\cdot2^4 + 1\cdot2^3 + 1\cdot2^1 + 1\cdot2^0$$
$$= 128 + 16 + 8 + 2 + 1$$

D → B (integer)

2) 13		
2) 6	1	
2) 3	0	
1	1	→ 1101_2

Ex. B ⟷ H

11 1110 1011 1100
3 E B C

Note. D → B (fraction)

Ex. 0.75

Modular Exponentiation

Motivation: $b^n \bmod m$ quickly

Procedure modular_expo(b, n = $(a_{k-1}, a_{k-2}, .. a_1 a_0)_2$, m)
  x:= 1
  power:= b mod m
  for i:= 0 to k-1 do
    if a_i= 1 then x:= (x·power) mod m endif
    power:= (power·power) mod m
  endfor
 // x equals b mod m //

Ex 11. Find $3^{644} \bmod 645$

Euclidian Algorithm
- **gcm**

Ex. gcd(91,287) //287 = 91·3 + 14 //
gcd(91,14) // 91 = 14·6 + 7 //
gcd(14,7) = 7

Note. $a = bq + r \rightarrow gcd(a,b) = gcd(b,r)$

```
procedure gcd(a,b)   // a > b //
   x:= a
   y:= b
   while y ≠ 0 do
      r:= x mod y
      x:= y
      y:= r
   endwhile
// x = gcd(a,b) //
```

Ex 12 . gcd(662,414)

	r = 248	r = 166	r = 82	r = 2	r = 0
x = 662	x = 414	x = 248	x = 166	x = 82	x = 2 ←
y = 414	y = 248	y = 166	y = 82	y = 2	y = 0