LEC 9 09/15/99

DIFFERENT BASES

THE EUCLIDEAN ALGORITHM OF FINDING g.c.d.

$gcd(45, 111) = ?$ GIVEN A PAIR $((45, 111))$

1. DIVIDE THE LARGER ONE BY THE SMALLER

$$111 = 2 \cdot 45 + 21$$

2. REPLACE THE LARGER BY THE REMAINDER

$(45, 111) \longrightarrow (21, 45)$ AND GOTO 1.

3. IF THE REMAINDER IS 0, THEN THE DIVISOR IS gcd.

$$45 = 21 \cdot 2 + 3 \qquad (3, 21)$$

$$21 = 7 \cdot \underset{\substack{\uparrow \\ gcd(45,111)}}{3} + \underline{\underline{0}}$$

LEMMA $a = bq + r \implies gcd(a,b) = gcd(b,r)$

PROOF. IT SUFFICES TO SHOW THAT

$$d \mid a \wedge d \mid b \iff d \mid b \wedge d \mid r \quad \Big\| \quad a = bq + r, \quad r = a - bq$$

Th. THE EUCLIDEAN ALGORITHM INDEED COMPUTES $gcd(a,b)$.

PROOF. $a \geq b > 0 \quad r_0 = a, \; r_1 = b$

$$r_0 = r_1 \cdot q_1 + r_2 \qquad 0 \leq r_2 < r_1$$

$$r_1 = r_2 \cdot q_2 + r_3 \qquad 0 \leq r_3 < r_2$$

$$\cdots\cdots\cdots\cdots$$

$$r_{n-2} = r_{n-1} \cdot q_{n-1} + r_n \qquad 0 \leq r_n < r_{n-1}$$

$$r_{n-1} = r_n \cdot q_n \qquad (*)$$

$$gcd(a,b) \underset{\text{RENAMING } a,b}{=} gcd(r_0, r_1) \underset{\text{BY LEMMA}}{=} gcd(r_1, r_2) = \ldots = gcd(r_{n-1}, r_n) \underset{\text{BY } (*)}{=}$$

PROCEDURE: gcd(a,b: POSITIVE INTEGERS)

```
x := a , y := b
WHILE y ≠ 0
BEGIN
     r := x mod y
     x := y
     y := r
```

DECIMAL NOTATION: $1999 = 1 \cdot 1000 + 9 \cdot 100 + 9 \cdot 10 + 9 =$

$$= 1 \cdot 10^3 + 9 \cdot 10^2 + 9 \cdot 10^1 + 9 \cdot 10^0$$

BINARY NOTATION: $(1101)_2 = 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 =$

$$= 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (13)_{10}$$

Th. LET $b > 1$ BE AN INTEGER. THEN EACH POSITIVE INTEGER n CAN BE EXPRESSED UNIQUELY IN THE FORM

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

WHERE $k, a_k, a_{k-1}, \ldots, a_1, a_0 \geq 0$

$a_i < b \quad (i = 0, 1, \ldots k), \quad a_k \neq 0$

PROOF. EXISTENCE - DIVIDING QUOTIENTS

$n = bq_0 + a_0$	$q_0 < n$	$n = b(bq_1 + a_1) + a_0 =$
$q_0 = bq_1 + a_1$	$q_1 < q_0$	$= q_1 b^2 + a_1 b + a_0 =$
$q_1 = bq_2 + a_2$	$q_2 < q_1$	$= (bq_2 + a_2) b^2 + a_1 b + a_0 =$
$\cdots$		$= q_2 b^3 + a_2 b^2 + a_1 b + a_0 =$
$q_{k-1} = b \cdot 0 + a_k$		$\cdots$
		$= a_k b^k + \cdots + a_2 b^2 + a_1 b + a_0$

UNIQUENESS - TAKING THE DIFFERENCE

$$n = \underbrace{a_k b^k + \cdots + a_1 b + a_0}_{n} = \underbrace{c_k b^k + \cdots + c_1 b + c_0}_{n}$$

$$n - n = 0$$

$$(a_k - c_k) b^k + \cdots + (a_1 - c_1) b + (a_0 - c_0) = 0$$

LET m = THE LARGEST i SUCH THAT $a_i - c_i \neq 0$ (IF ANY)

$$(a_m - c_m) \cdot b^m \overset{*}{=} (c_{m-1} - a_{m-1}) \cdot b^{m-1} + \cdots + (c_1 - a_1) b + (c_0 - a_0)$$

$$|(a_m - c_m) \cdot b^m| \geq b^m$$

$$|(c_{m-1} - a_{m-1}) \cdot b^{m-1} + \cdots + (c_1 - a_1) b + (c_0 - a_0)| \leq$$

$$\leq |c_{m-1} - a_{m-1}| \cdot b^{m-1} + \cdots + |c_1 - a_1| \cdot b + |c_0 - a_0| \leq$$

$$\leq (b-1) \cdot b^{m-1} + \cdots + (b-1) \cdot b + (b-1) =$$

$$= (b-1)(1 + b + b^2 + \cdots + b^{m-1}) = (b-1) \cdot \frac{b^m - 1}{b-1} = b^m - 1$$

THE EQUALITY $(*)$ IS IMPOSSIBLE $\implies a_i - c_i = 0$ FOR ALL $i = 0, 1, \ldots, k$. ■

UNIQUENESS IS BASED ON THE FACT THAT THE LARGEST m-DIGIT NUMBER IS LESS THAN THE SMALLEST $m+1$-DIGIT ONE: $(b-1) b^{m-1} \ldots (b-1) b + (b-1) < b^m$

EXAMPLE. FIND THE BASE 7 EXPANSION OF $(12345)_{10}$

$12345 = 7\cdot 1763 + 4$	$a_0 = 4$
$1763 = 7\cdot 251 + 6$	$a_1 = 6$
$251 = 7\cdot 35 + 6$	$a_2 = 6$
$35 = 7\cdot 5 + 0$	$a_3 = 0$
$5 = 7\cdot 0 + 5$	$a_4 = 5$

$$(12345)_{10} = (50664)_7$$

HEXADECIMAL EXPANSION — BASE 16

SIXTEEN DIGITS: 0123456789ABCDEF

(A = 10, B = 11, C = 12, D = 13, E = 14, F = 15)

$$(1A2B3C)_{16} = 1\cdot 16^5 + 10\cdot 16^4 + 2\cdot 16^3 + 11\cdot 16^2 + 3\cdot 16 + 12 =$$
$$= (1715004)_{10}$$

ADDITION OF INTEGERS (USUAL ALGORITHM)

$(10110)_2 + (11011)_2 = ?$

```
c:   1111      <- CARRY
a:    10110
b:    11011
     ------
s:   110001
```

$s_i = a_i + b_i + c_i \bmod 2$

$c_{i+1} = \lfloor (a_i + b_i + c_i)/2 \rfloor$

ADDITION ALGORITHM (PSEUDOCODE)

PROCEDURE add (a, b : positive integers)

{the binary expansions are $(a_{n-1}a_{n-2}\dots a_1a_0)_2$ and $(b_{n-1}b_{n-2}\dots b_1b_0)_2$ }

```
c := 0
FOR j := 0 TO n-1
BEGIN
    d := ⌊(a_j + b_j + c)/2⌋
    s_j := a_j + b_j + c - 2d
    c := d
END
s_n := c
```

{the binary expansion of the sum is $(s_n s_{n-1} \dots s_1 s_0)_2$ }

COMPLEXITY OF THIS ALGORITHM (NUMBER OF BIT ADDITIONS): ≤ 3 additions at each step, n steps $= O(n)$ additions

MULTIPLICATION OF BINARY EXPANSIONS

```
   × { 1011        a          a = (11)_10
     { 1101        b          b = (13)_10
     -------
  +  {    1011     c_0        ab = (143)_10
     {   0000      c_1
     {  1011       c_2
     { 1011        c_3
     ----------
      10001111     PRODUCT a·b
      1111         <- CARRY
```

PROCEDURE multiply (a, b: positive integers)

{ $a = (a_{n-1}a_{n-2}\dots a_1a_0)_2$, $b = (b_{n-1}b_{n-2}\dots b_1b_0)_2$ }

```
FOR j := 0 TO n-1
BEGIN
    IF b_j = 1 THEN c_j := a shifted j places
    ELSE c_j := 0
END
```

{ $c_0, c_1, c_2 \dots c_{n-1}$ ARE THE PARTIAL PRODUCTS }

```
p := 0
FOR j := 0 TO n-1
    p := p + c_j        {p IS THE VALUE OF a·b}
```

COMPLEXITY (SHIFTS AND ADDITIONS OF BITS)

SH: $0 + 1 + 2 + \dots + (n-1) = O(n^2)$; AD: $(n-1)\cdot O(n) = O(n^2)$

TOTAL: $O(n^2)$

HW 2.4: 2f 8b,d 32e 36