Un algoritme de divisió a l'antic Egipte

Ramon Nolla
IES Pons d'Icart

Gener/2002

Suposem que es vol dividir 511 entre 23. És a dir, cerquem dos nombres $q, r \in \mathbb{N}$, tals que $511 = 23 \cdot q + r$ i $0 \leq r < b$. L'algoritme egipci consisteix en:

2^i	$2^i \cdot 23$
1	23
2	46
4	92
8	184
16	368

1) Construir una taula de totes les parelles $< 2^i \, , \, 2^i \cdot 23 >$, $i \in \mathbb{N}$, tals que $2^i \cdot 23 \leq 511$

2) Cercar la millor aproximació per defecte, del dividend 511, proporcionada per la suma de nombres adequats de la columna $2^i \cdot 23$ de la taula. Per aconseguir-ho s'actua de la manera següent:

- A l'últim element 368 se li suma l'anterior 184. Llavors:
  (a) Si, com en el cas present, el resultat $368 + 184 = 552$ és major que el dividend 511, es prescindeix del terme 184 afegit. Llavors, 368 és la primera aproximació del dividend 511.
  (b) Si, el resultat 552, hagués sigut igual que el dividend 511, la millor aproximació s'hauria trobat.
  (c) Si, el resultat 552, hagués sigut menor que el dividend 511, es consideraria 552 com a primera aproximació.
- Aquesta operació es repeteix, en els casos (a) i (c), sobre la primera aproximació obtinguda, sumant-li l'element anterior 92. S'actua igual sobre les successives aproximacions fins arribar al cap de dalt de la taula o obtenir el dividend 511. D'aquesta manera, en el nostre cas, s'obté la millor aproximació $368 + 92 + 46 = 506$.

3) Calcular:

- El quocient, mitjançant la suma $2 + 4 + 16 = 22$ dels termes 2, 4, 16 aparellats amb els elements 46, 92, 368 que han proporcionat la millor aproximació.
- El residu, mitjançant la diferència $511 - 506 = 511 - 23 \cdot 22 = 5$ entre el dividend i l'aproximació obtinguda.

2^i		$2^i \cdot 23$
1		23
$\boxed{2}$	$\longleftarrow$	$\boxed{46}$
$\boxed{4}$	$\longleftarrow$	$\boxed{92}$
8		184
$\boxed{16}$	$\longleftarrow$	$\boxed{368}$
22		506

1 Programació del procediment

Plantejament general

Donats dos nombres $a, b \in \mathbb{N}$, $b \neq 0$, descomponem el procediment per a trobar el quocient i el residu de la divisió de a entre b en tres etapes:

1. Construcció d'una taula amb totes les parelles $< 2^i \, , \, 2^i \cdot b >$, $i \in \mathbb{N}$, tals que $2^i \cdot b \leq a$.
2. Recerca de la millor aproximació del dividend a proporcionada per la suma de nombres $2^i \cdot b$ adequats de la taula.

3. Si aquesta aproximació és $2^{\alpha_1} \cdot b + 2^{\alpha_2} \cdot b + \cdots 2^{\alpha_p} \cdot b$, s'escriu:
   – Quocient $= 2^{\alpha_1} + 2^{\alpha_2} + \cdots 2^{\alpha_p}$.
   – Residu $= a - (2^{\alpha_1} \cdot b + 2^{\alpha_2} \cdot b + \cdots 2^{\alpha_p} \cdot b)$.

Programació

El programa serà elaborat en dues fases:

a) Definició de la funció auxiliar TDEE(a, b) que proporcionarà la taula de l'etapa 1.

b) Definició de la funció DEE(a, b) que proporcionarà el quocient i el residu de a entre b, a partir de les accions desenvolupades en les etapes 2 i 3.

Funció TDEE(a, b)

Donats dos nombres $a, b \in \mathbb{N}$, $b \neq 0$, es tracta de construir una taula de totes les parelles $< 2^i \ , \ 2^i \cdot b >$, $i \in \mathbb{N}$, tals que $2^i \cdot b \leq a$. Per aconseguir-ho, emmagatzemarem els valors dels successius valors de 2^i i $2^i \cdot b$ en dos vectors:

$$d := [2^0, 2^1, 2, 2^2, \ldots, 2^i] \quad \text{i} \quad w := [2^0 b, 2^1 b, 2^2 b, \ldots, 2^i \cdot b].$$

El procediment, que anomenarem TDEE(a, b), consistirà en:

1) Fer un test de veritat o falsedat de l'enunciat

$$k \cdot b > a, \quad \text{en què} \quad k = 2^i.$$

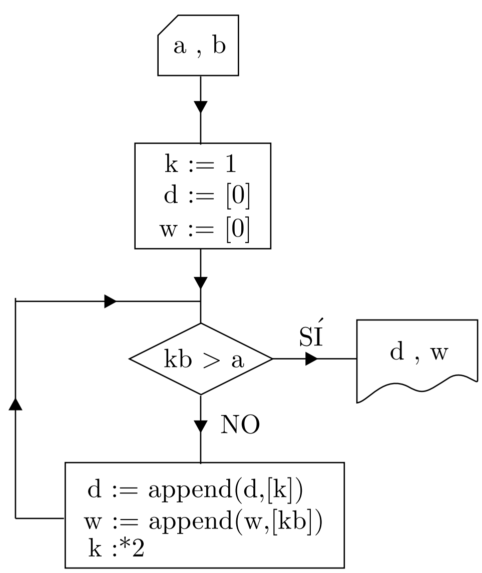
2) Segons el resultat, actuar com indiquem a l'esquema següent:

– **Si és veritat:** Escriure els vectors d i w, els quals constitueixen la taula cercada, i s'ha acabat.

– **Si és fals:** Guardar k i $k \cdot b$ com a noves coordenades dels vectors d i w, respectivament. Després, multiplicar k per 2, i tornar a fer el test.

De tot això es conclou que caldrà inicialitzar tres variables: k numèrica, d i w vectorials. Les variables d i w s'inicialitzaran a [0] perquè serà útil per al cas $a < b$, en què el quocient és 0.

TDEE(a,b) :=



Funció DEE(a, b)

El procediment per trobar el quocient i el residu, que anomenarem DEE(a, b), consistirà en:

1) Considerar les successives sumes

$$y := 2^{\alpha_1} \cdot b + 2^{\alpha_2} \cdot b + \cdots 2^{\alpha_p} \cdot b \quad \text{i} \quad x := 2^{\alpha_1} + 2^{\alpha_2} + \cdots 2^{\alpha_p} \,,$$

les últimes de les quals proporcionaran el quocient x i el residu $a - y$.

2) Sotmetre a un test de veritat l'enunciat

$$y = a \quad \vee \quad n = 1,$$

en què n és el lloc que ocupa en la seva columna de la taula, l'últim element $2^i \cdot b$ que s'ha sumat per aproximar a.

3) Observar el resultat del test:

Si és veritat: Escriure el valors del quocient x i del residu $a - y$. Això és així perquè, en ser la divisió exacta (cas $y = a$) o haver arribat al cap de dalt de la taula (cas $n = 1$), el procediment s'ha acabat.

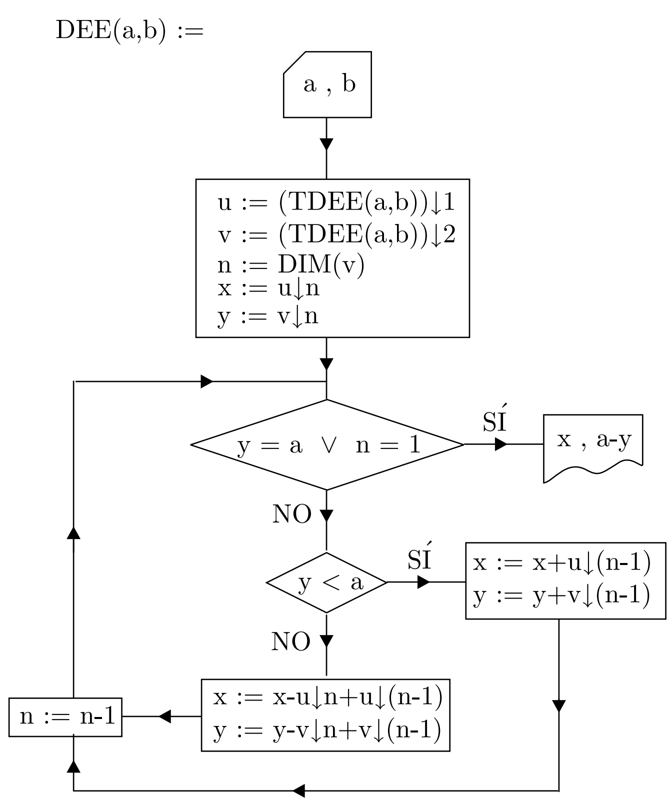
Si és fals: Sotmetre a test l'enunciat

$$y < a.$$

– Si aquest últim és veritat, sumar a y i a x els termes de la taula, anteriors als últims considerats.
– Si és fals, restar de y i de x els últims termes de la taula considerats —perquè $y > a$—, i sumar els termes de la taula anteriors a aquests.

4) Finalment, actualitzar l'últim lloc $n := n - 1$ de la taula que hem sumat, i tornar a fer el test del primer apartat.

Notem que caldrà inicialitzar cinc variables:
– u, v vectorials per a les columnes de la taula.
– n per al lloc que ocupen els elements de la taula que sumem.
– x, y per als successius valors de les sumes de les columnes de la taula.



2 Edició del programa

Per implementar aquest algoritme introduirem a la línia d'edició les instruccions següents:

- `TDEE(a, b) := PROG(k := 1, d := [0], w := [0], LOOP(IF(k·b > a, RETURN [d, w]), w := APPEND(w, [k·b]), d := APPEND(d, [k]), k :* 2))`

- `DEE(a, b) := PROG(IF(¬ INTEGER?(a) ∨ ¬ INTEGER?(b) ∨ a < 0 ∨ b ≤ 0, RETURN ‘‘Introdueix a i b naturals, b≠0 ’’), u := (TDEE(a, b))↓1, v := (TDEE(a, b))↓2, n := DIMENSION(v), x := u↓n, y := v↓n, LOOP(IF(y = a ∨ n = 1, RETURN [‘‘Quocient =’’, x; ‘‘Residu =’’, a - y]), IF(y < a, PROG(y := y + v↓(n-1), x := x + u↓(n-1)), PROG(y := y - v↓n + v↓(n-1), x := x - u↓n + u↓(n-1))), n := n - 1))`

Es pot observar que s'ha introduït un test per verificar si els nombres a i b a introduir són naturals o no, i si $b \neq 0$. A la finestra d'àlgebra el programa es visualitzarà així:

- TDEE(a, b) :=

```
      Prog
          k := 1
          d := [0]
          w := [0]
          Loop
              If k·b > a
                    RETURN [d, w]
              w := APPEND(w, [k·b])
              d := APPEND(d, [k])
              k :* 2
```

- DEE(a, b) :=

```
      Prog
          If ¬ INTEGER?(a) ∨ ¬ INTEGER?(b) ∨ a < 0 ∨ b ≤ 0
                RETURN Introdueix a i b naturals, b≠0
          u := (TDEE(a, b))↓1
          v := (TDEE(a, b))↓2
          n := DIMENSION(v)
          x := u↓n
          y := v↓n
          Loop
              If y = a ∨ n = 1
                    RETURN [ Quocient =, x; Residu =, a - y]
              If y < a
                    Prog
                        y := y + v↓(n - 1)
                        x := x + u↓(n - 1)
                    Prog
                        y := y - v↓n + v↓(n - 1)
                        x := x - u↓n + u↓(n - 1)
              n := n - 1
```