

Cap.1. Descrierea și evaluarea algoritmilor

Prof.dr.ing. Gabriela Ciuprina

Universitatea Politehnică București, Facultatea de Inginerie Electrică,
Departamentul de Electrotehnică

Suport didactic pentru disciplina *Metode numerice în ingineria electrică, 2017-2018*

Notes

Cuprins

- 1 Pseudocodul
 - Definiții
 - Sintaxa declarațiilor
 - Sintaxa instrucțiunilor

- 2 Complexitatea algoritmilor
 - Timp de calcul
 - Necesari de memorie

Notes

Algoritm

Metodă de rezolvare a unei probleme bazată pe descompunerea în etape simple, elementare.

Pseudocod

Metodă de descriere a algoritmilor.

- fără sintaxă strictă;
- clar, neambiguu;
- cuvinte cheie (în limba română).

Alcătuit din doua feluri de linii

- declarații - descriu datele;
- instrucțiuni - descriu acțiunile.

Notes

Tipuri de date

Fundamentale (simple)

logic, întreg, real, caracter

logic T, F ; adevărat (*true* - în engl.), fals
întreg N ; numărul de noduri
real Pc, Pg ; putere consumată, putere generată
caracter c, C

Agregate

tablou, înregistrare

tablou real $V[10]$

întreg N

tablou real $V[N]$

$V(1), V(5)$ sau
 V_1, V_5

Notes

Tipuri de date

Fundamentale (simple)

logic, întreg, real, caracter

logic T, F ; adevărat (*true* - în engl.), fals
întreg N ; numărul de noduri
real Pc, Pg ; putere consumată, putere generată
caracter c, C

Agregate

tablou, înregistrare

înregistrare punct punct.polar
logic polar punct.coord1
real coord1 punct.coord2
real coord2

Notes

Instrucțiuni de intrare/ieșire

citește N
citește q, p ; se admite citirea unei liste de variabile
scrie N
scrie q, p ; se admite scrierea unei liste de variabile

Notes

Instrucțiunea de atribuire

logic a ; rezultatul evaluării expresiei logice
logic b, c ; operanzi logici
real x, y ; operanzi aritmetici
 $a = (b \text{ sau } (\text{nu } c))$; expresie logică cu operatori logici
 $a = (x \leq y)$; expresie logică cu operatori de relație
 $a = (x = y)$; expresie logică cu operatori de relație

întreg i
real d, x, y
 $i = i + 1$
 $d = xy + \sin(y)$
 $d = \sqrt{d}$
 $d = \frac{\frac{d^2 + d}{2} + \frac{d}{3}}{\frac{1}{d}}$

Notes

Decizia fără alternativă

real x
 $x = 2$
dacă $x < 0$
 $x = x + 3$
 $x = x/2$
 $x = x^2$
 $x = 2x$

real x
 $x = 2$
dacă $x < 0$
 $x = x + 3$
 $x = x/2$
 $x = x^2$
 $x = 2x$

real x
 $x = 2$
dacă $x < 0$
 $x = x + 3$
 $x = x/2$
 $x = x^2$
 •
 $x = 2x$

Notes

Decizia cu alternativă

```
real x           ; un număr real - dată de intrare  
real modul      ; dată de ieșire - modulul numărului real  
citește x  
dacă x ≥ 0  
    modul = x  
altfel  
    modul = -x  
•  
scrie modul
```

Notes

Ciclul cu test initial

```
întreg N  
întreg i  
tablou real x[N]  
N = 3  
x1 = 1  
x2 = -1  
x3 = 2  
i = 1  
cât timp (i ≥ 1) și (i ≤ N)  
    dacă xi ≥ 0  
        scrie "Elementul ", i, " este pozitiv"  
    •  
    i = i + 1  
•
```

Notes

Ciclul cu test final

```
întreg  $N$   
tablou real  $x[N]$   
real  $s, \varepsilon$   
întreg  $k$   
...  
 $k = 0$   
 $s = 0$   
repetă  
     $k = k + 1$   
     $s = s + x_k$   
cât timp  $|x_k| \geq \varepsilon$ 
```

Notes

Ciclul cu test final

```
întreg  $N$   
tablou real  $x[N]$   
real  $s, \varepsilon$   
întreg  $k$   
...  
 $k = 0$   
 $s = 0$   
repetă  
     $k = k + 1$   
     $s = s + x_k$   
până când  $|x_k| < \varepsilon$ 
```

Notes

Ciclul cu contor

```
pentru contor = val_in, val_fin[, pas] [repetă]  
    secvență
```

```
întreg  $N$   
citește  $N$   
tablou real  $a[N, N]$   
întreg  $i, j$   
pentru  $i = 1, N$   
    pentru  $j = 1, N$   
        citește  $a_{i,j}$ 
```

Rutine: proceduri

Definiție:

```
procedură nume_proc (lista argumentelor formale de I/O)  
; comentarii ce descriu ce face procedura și parametrii acesteia  
...  
; declarații pentru argumente  
...  
; instrucțiuni  
...  
retur ; se comandă întoarcerea în punctul de apel
```

Apel:

```
nume_proc (lista argumentelor actuale de intrare și ieșire)
```

Notes

Notes

Rutine: funcții

Definiție:

```
funcție nume_fct (lista argumentelor formale de intrare)  
; comentarii ce descriu ce face funcția și parametrii acesteia  
...  
; declarații pentru argumente  
...  
; instrucțiuni  
...  
întoarce valoare ; se comandă întoarcerea în punctul de apel
```

Apel:

```
val = nume_fct (lista argumentelor actuale de intrare)
```

Exemplu

```
; program principal  
întreg N  
citește N  
tablou real a[N], b[N]  
real p  
citește_vector(N,a)  
citește_vector(N,b)  
p = produs_scalar(N,a,b)  
scrie p  
  
procedură citește_vector(N,x)  
întreg N  
tablou real x[N]  
întreg i  
pentru i = 1, N  
    citește xi  
•  
  
funcție produs_scalar(N,v,w)  
întreg N  
tablou real v[N], w[N]  
întreg i  
real r  
r = 0  
pentru i = 1, N  
    r = r + viwi  
•  
întoarce r
```

Notes

Notes

T = O(?)

Complexitatea unui algoritm din punct de vedere al timpului de calcul

relația dintre timpul de calcul exprimat în număr de operații elementare și dimensiunea problemei

Operație elementară

operația care durează cel mai mult.

Algoritmi polinomiali - de ordinul 1 (liniar)

p = 0;

pentru $i = 1, n$

$p = p + a_i b_i$

$T = O(2n) \approx O(n)$

•

Notes

Notes

Algoritmi polinomiali - de ordinul 2 (pătratic)

```
pentru i = 1, n
  bi = 0
  pentru j = 1, n
    bi = bi + aijxj
  •
•
```

$T = O(2n^2) \approx O(n^2)$

Notes

Algoritmi polinomiali - de ordinul 3 (cubic)

```
pentru i = 1, n
  pentru j = 1, n
    cij = 0
    pentru k = 1, n
      cij = cij + aikbkj
    •
  •
•
```

$T = O(2n^3) \approx O(n^3)$

Notes

Algoritmi polinomiali - de ordinul k

$$T = O(n^k) \iff (\exists) C > 0 \text{ și } n_0 \text{ a.i. } T \leq Cn^k (\forall) n \geq n_0$$

Algoritm de ordin 0: $T = O(1)$ - nu depinde de dimensiunea problemei.

$$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < \dots < O(e^n) < O(n!)$$

Notes

Ex1: Evaluarea unui polinom - varianta I

$$P(x) = a_0 + a_1x + \dots + a_nx^n. \quad (1)$$

$$P(x) = a_0 + \sum_{i=1}^n a_i x^i \quad (2)$$

; Varianta 1

$$P = a_0;$$

pentru $i = 1, n$

$$t = a_i$$

pentru $j = 1, i$

$$t = t * x$$

$$P = P + t;$$

•

Nr. de operații elementare:

$$\sum_{i=1}^n (i + 1) \approx n^2 / 2$$

$$T_1 = O(n^2 / 2) \approx O(n^2)$$

Notes

Ex1: Evaluarea unui polinom - varianta II

$$P(x) = a_0 + a_1 t_1 + a_2 t_2 + \dots + a_n t_n = a_0 + \sum_{i=1}^n a_i t_i, \quad (3)$$

unde

$$t_1 = x, \quad t_i = t_{i-1} x, \quad i = 2, \dots, n. \quad (4)$$

; Varianta 2

$$P = a_0;$$

$$t = 1;$$

$$\text{pentru } i = 1, n \quad T_2 = O(3n) \approx O(n)$$

$$t = t * x$$

$$P = P + a_i * t$$

•

Ex1: Evaluarea unui polinom - varianta III

$$P(x) = a_0 + x(a_1 + x(a_2 + \dots x(a_{n-1} + a_n x) \dots)). \quad (5)$$

; Varianta 3

$$P = a_n;$$

$$\text{pentru } i = n - 1, 0, -1 \quad T_3 = O(2n) \approx O(n)$$

$$P = a_i + P * x$$

•

Notes

Notes

Ex2: Reducerea timpului de calcul

```
; Varianta A
întreg  $i, j, n$ 
real  $a, b$ 
...
tablou real  $c[n][n]$ 
pentru  $i = 1, n$ 
    pentru  $j = 1, n$ 
         $c_{ij} = f(i * a) + f(j * b)$  ;  $f$  definită în altă parte
    •
•
```

Operație elementară: evaluarea funcției $f \Rightarrow T_A = O(2n^2)$.

Notes

Ex2: Reducerea timpului de calcul

```
; Varianta B
întreg  $i, j, n$ 
real  $a, b, p$ 
...
tablou real  $c[n][n]$ 
pentru  $i = 1, n$ 
     $p = f(i * a)$ 
    pentru  $j = 1, n$ 
         $c_{ij} = p + f(j * b)$ 
    •
•
```

$T_B = O(n(n + 1)) = O(n^2 + n) \approx O(n^2)$.

Notes

Ex2: Reducerea timpului de calcul

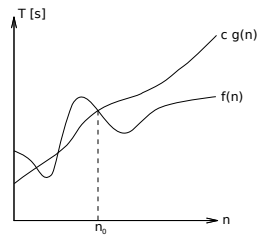
```

; Varianta C
întreg i, j, n
real a, b
...
tablou real c[n][n]
tablou real p[n], q[n]
pentru i = 1, n
    pi = f(i * a)
    qi = f(i * b)
    •
    pentru i = 1, n
        pentru j = 1, n
            cij = pi + qj
        •
    •
    
```

$T_C = O(2n)$

Notes

Notăție asimptotică O



$$T = f(n) = O(g(n))$$

$$T = O(g(n)) \iff \exists C > 0 \text{ și } n_0 \text{ a.i. } T \leq Cg(n) \text{ (}\forall n \geq n_0\text{)}$$

Notes

M = O(?)

Complexitatea unui algoritm din punct de vedere al necesarului de memorie

relația dintre necesarul de memorie exprimat în număr de locații elementare și dimensiunea problemei

Locația elementară de memorie

locația corespunzătoare unui număr real.

$$M = O(n^k) \iff (\exists) C \text{ a.i. } M \leq Cn^k.$$

Ex2: Reducerea timpului de calcul

; Varianta A

întreg i, j, n

real a, b

...

tablou real $c[n][n]$

pentru $i = 1, n$

pentru $j = 1, n$

$c_{ij} = f(i * a) + f(j * b)$; f definită în altă parte

-
-

$$T_A = O(2n^2)$$

$$M_A = O(n^2 + 2) \approx O(n^2)$$

Notes

Notes

Ex2: Reducerea timpului de calcul

; Varianta B

întreg i, j, n

real a, b, p

...

tablou real $c[n][n]$

pentru $i = 1, n$

$p = f(i * a)$

pentru $j = 1, n$

$c_{ij} = p + f(j * b)$

-
-

$$T_B = O(n(n+1)) = O(n^2 + n) \approx O(n^2)$$

$$M_B = O(n^2 + 3) \approx O(n^2)$$

Notes

Ex2: Reducerea timpului de calcul

; Varianta C

întreg i, j, n

real a, b

...

tablou real $c[n][n]$

tablou real $p[n], q[n]$

pentru $i = 1, n$

$p_i = f(i * a)$

$q_i = f(i * b)$

$$T_C = O(2n)$$

$$M_C = O(n^2 + 2n + 2) \approx O(n^2)$$

-
-

pentru $i = 1, n$

pentru $j = 1, n$

$c_{ij} = p_i + q_j$

-
-

Notes

Concluzii

- Elaborarea unui algoritm nu se face în fața calculatorului. De aceea este nevoie de un pseudolimbaj (pseudocod).
- Pseudocodul prezentat este cel pe care îl veți întâlni în îndrumar. Puteți să îl alterați ușor, fără a fi penalizați, dacă el rămâne clar și neambiguu.
- **Elaborarea unui algoritm înseamnă întotdeauna stabilirea unui compromis între timp de calcul și necesar de memorie.**

Temă - pentru bonus

- Propuneți o structură de date care să descrie un circuit rezistiv liniar de curent continuu. (Indicație: amintiți-vă cum erau descrise astfel de circuite în limbaj Spice).
- Scrieți pseudocodul unei proceduri care să aibă ca argument de intrare numele unui fișier de tip Spice și ca argument de ieșire structura de date pe care ați conceput-o anterior.
- Evaluați complexitatea pseudocodului din punct de vedere al necesarului de memorie și din punct de vedere al timpului de calcul, alegându-vă în mod potrivit: unul sau mai mulți parametri de care depinde complexitatea procedurii și operația de referință.

Tema nu se va redacta electronic. Tema se va preda la curs. Vor primi bonus pe ea primii 5 studenți care vor preda o rezolvare completă, corectă în proporție de mai mult de 80 %.

Notes

Notes
