Structuri de date și algoritmi
Complexitate

Lect. Dr. Ing. Cristian Chilipirea – cristian.chilipirea@mta.ro

Un număr nu e niciodată suficient!!!!

Unitatea de măsură este o metodă implicită de comparație.

Măsurarea timpului de execuție

time ./executabil p a r a m e t r i

Măsurare timp – Linia de comandă

time *sleep 5*

real 0m5.001s
user 0m0.000s
sys 0m0.001s

time sleep 5

sleep 5 0.00s user 0.00s system 0% cpu 5.002 total

/usr/bin/time sleep 5

0.00user 0.00system 0:05.00elapsed 0%CPU (0avgtext+0avgdata 2076maxresident)k
0inputs+0outputs (0major+73minor)pagefaults 0swaps

Măsurare timp – Linia de comandă

***time** sleep 5*

real 0m5.001s
user 0m0.000s
sys 0m0.001s

Wall clock time – Timpul trecut de la pornirea programului – Pe acesta îl folosim

time sleep 5

sleep 5 0.00s user 0.00s system 0% cpu 5.002 total

/usr/bin/time sleep 5

0.00user 0.00system 0:05.00elapsed 0%CPU (0avgtext+0avgdata 2076maxresident)k
0inputs+0outputs (0major+73minor)pagefaults 0swaps

Măsurare timp – Linia de comandă

time sleep 2

real 0m2.0**21**s
user 0m0.000s
sys 0m0.000s

time sleep 2

real 0m2.0**18**s
user 0m0.000s
sys 0m0.016s

Timpii măsurați nu sunt exacți. Pentru a măsura corect trebuie să facem medie a timpilor după mai multe rulări sau să considerăm doar timpi mari – peste o secundă.

time sleep 2

real 0m2.0**16**s
user 0m0.000s
sys 0m0.000s

time sleep 2

real 0m2.0**15**s
user 0m0.000s
sys 0m0.000s

Măsurare timp – Linia de comandă

***time** sleep 5*

real 0m5.001s
user 0m0.000s
sys 0m0.001s

Suma timpului petrecut în user space pe fiecare core.

time sleep 5

sleep 5 0.00s user 0.00s system 0% cpu 5.002 total

/usr/bin/time sleep 5

0.00user 0.00system 0:05.00elapsed 0%CPU (0avgtext+0avgdata 2076maxresident)k
0inputs+0outputs (0major+73minor)pagefaults 0swaps

Măsurare timp – Linia de comandă

***time** sleep 5*

real 0m5.001s
user 0m0.000s
sys 0m0.001s

Suma timpului petrecut în kernel pe fiecare core.

time sleep 5

sleep 5 0.00s user 0.00s system 0% cpu 5.002 total

/usr/bin/time sleep 5

0.00user 0.00system 0:05.00elapsed 0%CPU (0avgtext+0avgdata 2076maxresident)k
0inputs+0outputs (0major+73minor)pagefaults 0swaps

Măsurare timp – Linia de comandă

Operațiile de I/O sunt executate de Kernel

```
time dd if=/dev/zero of=file.txt count=1024 bs=1    048576
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 9.4847 s, 113 MB/s

real    0m9.490s
user    0m0.000s
sys     0m0.992s
```

Măsurare timp cu sau fără I/O?

Mașina Turing

Alan Turing 1912-1954

ON COMPUTABLE NUMBERS, WITH AN APPLICATION TO THE ENTSCHEIDUNGSPROBLEM

By A. M. Turing.

[Received 28 May, 1936.—Read 12 November, 1936.]

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. Although the subject of this paper is ostensibly the computable *numbers*, it is almost equally easy to define and investigate computable functions ...able or a real or computable variable, computable ... forth. The fundamental problems involved are, ... each case, and I have chosen the computable numbers ...nt as involving the least cumbrous technique. I hope ...account of the relations of the computable numbers, ...rth to one another. This will include a development ...nctions of a real variable expressed in terms of com- ... According to my definition, a number is computable ... written down by a machine.

Calcul timp execuție

```
1  int sum(int *v, int n)
2  {
3      int s = 0;
4      int i = 0;
5      while (i < n)
6      {
7          s += v[i];
8          i++;
9      }
10     return s;
11 }
```

Linia	Cost	Repetiții
3	C1 (atribuire)	1
4	C1 (atribuire)	1
5	C2 (comp + jump)	n + 1
7	C3 (adunare + adunare pointer + load)	n
8	C4 (incrementare)	n

Timpul total de executie :

$$T(n) = 2C_1 + (n+1)C_2 + C_3 n + C_4 n$$
$$= (2C_1 + C_2) + \boldsymbol{n} * (\boldsymbol{C_2} + \boldsymbol{C_3} + \boldsymbol{C_4})$$
$$3C_{min} + 3nC_{min} \leq T(n) \leq 3C_{max} + 3nC_{max}$$

Calcul timp execuție

Simplificare :

- Operaţiile fundamentale au acelaşi cost unitar (1)

Timpul total de execuţie :

$$T(n) = (2C_1 + C_2) + \boldsymbol{n} * (\boldsymbol{C_2} + \boldsymbol{C_3} + \boldsymbol{C_4})$$
$$= 3 + 3n = \Theta(n) \text{ (funcţie liniară)}$$

- Constantele au o importanţă relativă ; o constantă poate fi cumulul unor operaţii (ex. cazul buclei for)
- **Algoritmul are un timp de execuţie (ordin) liniar!**

Calcul timp execuție

```
1  int search(int *v, int n, int a)
2  {
3      int i = 0;
4      while (i < n)
5      {
6          if (v[i] == a)
7          {
8              return i;
9          }
10         i++;
11     }
12     return -1;
13 }
```

Linia	Cost	Repetiții
3	C1 (atribuire)	1
4	C2 (comp + jump)	σ(a)
6	C3 (test egalitate)	σ(a)
10	C4 (incrementare)	σ(a)-1

$\sigma(a)$ = pozitia lui a în cadrul vectorului (valori între 1 și n)

Timpul total de execuție (a există în vector):

$$T(n) = C_1 - C_4 + (C_2+C_3+C_4)*\sigma(a) = 3\sigma(a)$$

Timpul total de execuție (a nu există în vector):

$$T(n) = C_1 + C_2(n+1)+(C_3+C_4)n = 3n+2$$

Calcul timp execuție

Cazul valorii a în vectorul v : $T(n) = 3\sigma\ (a)$

Cazul în care a nu există în vector : $T(n) = 3n+2$

- Cazul cel mai favorabil : $T(n) = 3 = \Theta(1)$
- Cazul cel mai defavorabil : $T(n) = 3n+2 = \Theta(n)$
- Cazul mediu : depinde de distribuţia probabilistică a intrărilor: $\sigma(a)$ este o variabilă aleatoare (1 … n)

Calcul timp execuție

Cazul mediu : depinde de distribuţia probabilistică a intrărilor: $\sigma(a)$ este o variabilă aleatoare $(1 \dots n)$

Distribuţia celor N cazuri posibile : $\begin{pmatrix} T_1 & \dots & T_N \\ p_1 & \dots & p_N \end{pmatrix}$

$$\boldsymbol{T_{mediu}(n)} = \sum p_i * T_i =$$

$$\frac{3n+2}{n+1} + \sum \frac{1}{n+1} * 3 * \sigma(i) = \frac{3n+2}{n+1} + \frac{\boldsymbol{3n}}{\boldsymbol{2}}$$

Ordinul sau Rata de Creştere

Timpul de execuţie : T(n)

- $T(n) = an^2 + bn + c$ sau $T(n) = qn + r$
- Constantele care contează sunt cele care însoţesc factorul dominant (an^2 sau qn)
- Constantele factorului dominant contează mai puţin decât ordinul acestuia: pătratic, liniar, logaritmic, etc.

Ordinul sau rata de creştere

- Cum creşte timpul de execuţie dacă dimensiunea datelor de intrare se dublează/triplează ?
- $an^2 + bn + c = \Theta(n^2)$ sau $T(n) = qn + r = \Theta(n)$

Notaţia O

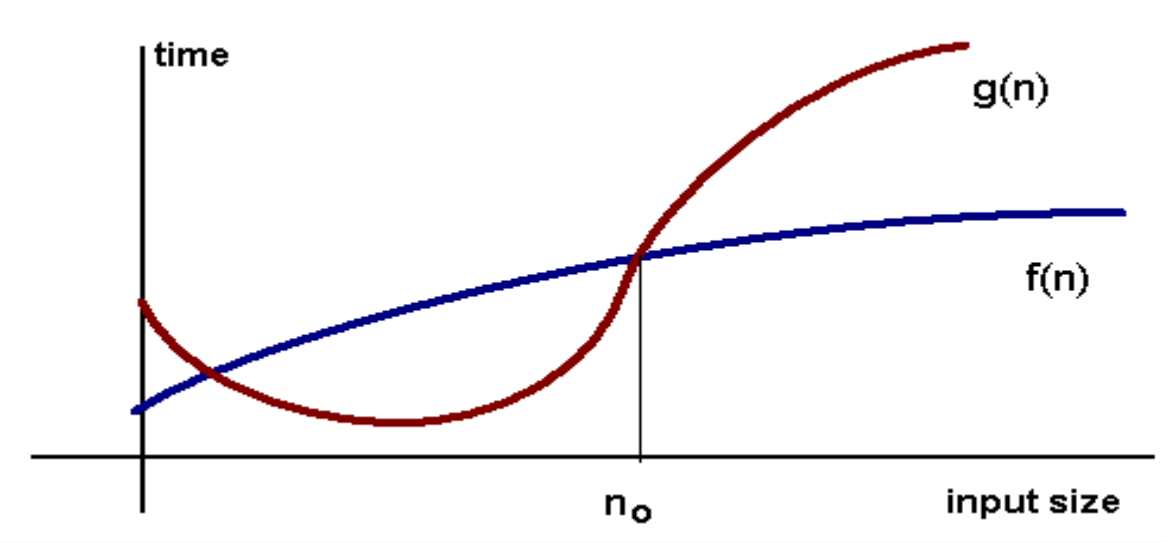
- Ordin de creştere pentru cazul cel mai defavorabil

Definitia formală a notaţiei O

Fie două functii $f: N \to N$ si g: $N \to N$. $f \in O(g)$:

$\exists\, n_0 \in N, c > 0$ astfel încât $\forall n \geq n_0, f(n) \leq c * g(n)$

Exemple : $5n^2 + 3n + 1 \in O(n^2)$, $3n + \ln(n) \in O(n)$



Notaţiile asimptotice Ω şi θ

Cazul cel mai favorabil pentru timpul de execuţie

Definiţia formală a notaţiei Ω

Fie două funcţii $f: N \to N$ şi g: $N \to N$. $f \in \Omega(g)$:

$\exists\, n_0 \in N, c > 0$ astfel încât $\forall n \geq n_0, f(n) \geq c * g(n)$

Exemple : $5n^2 + 3n + 1 \in \Omega(n^2)$, $3n + \ln(n) \in \Omega(n)$

Definiţia formală a notaţiei θ

Fie două funcţii $f: N \to N$ şi g: $N \to N$. $f \in \Theta(g)$:

$\exists\, n_0 \in N, c_1, c_2 > 0$ astfel încât $\forall n \geq n_0$:

$c_1 * g(n) \leq f(n) \leq c_2 * g(n)$ sau $\lim_{n\to\infty} \frac{f(n)}{g(n)} = ct.$

Notaţii complexitate

O – ordin creștere caz cel mai defavorabil

Fie două functii $f: N \to N$ si g: $N \to N$. $f \in O(g)$:

$\exists\, n_0 \in N, c > 0$ a.î. $\forall n \geq n_0, f(n) \leq c * g(n)$

Θ – ordin de creștere cel mai apropiat (și favorabil și defavorabil)

Fie două funcţii $f: N \to N$ şi g: $N \to N$. $f \in \Theta(g)$:

$\exists\, n_0 \in N, c_1, c_2 > 0$ astfel încât $\forall n \geq n_0$

$c_1 * g(n) \leq f(n) \leq c_2 * g(n)$ sau $\lim_{n \to \infty} \frac{f(n)}{g(n)} = ct.$

Ω – ordin creștere caz cel mai favorabil

Fie două funcţii $f: N \to N$ şi g: $N \to N$. $f \in \Omega(g)$:

$\exists\, n_0 \in N, c > 0$ astfel încât $\forall n \geq n_0, f(n) \geq c * g(n)$

Confuzia O, Ω şi θ

Ω limită inferioară

Θ limită inferioară și superioară

O limită superioară

În industrie toata lumea folosește notația O când defapt se referă la θ.

Proprietăţi de calcul ale notaţiei θ

$\Theta(c * f) = \Theta(f)$ oricare ar fi constanta c

$\Theta(f) + \Theta(g) = \Theta(f + g) = \max(\Theta(f), \Theta(g))$

$\Theta(f) * \Theta(g) = \Theta(f * g)$ (ex. bucle imbricate)

Notație	Referință	10^x	Aprox 2^x
p	0.000_000_000_001	10^{-12}	2^{-40}
n	0.000_000_001	10^{-9}	2^{-30}
u	0.000_001	10^{-6}	2^{-20}
m	0.001	10^{-3}	2^{-10}
	1	10^{0}	2^{0}
k	1000	10^{3}	2^{10}
M	1000_000	10^{6}	2^{20}
G	1000_000_000	10^{9}	2^{30}
T	1000_000_000_000	10^{12}	2^{40}
P	1000_000_000_000_000	10^{15}	2^{50}
E	1000_000_000_000_000_000	10^{18}	2^{60}

Timp

UMI	Secunde
ns	0.000000001
us	0.000001
ms	0.001
	1
m	60s
h	3600s
day	86400s
year	31190400s

Timpi Execuție

Complexitate	Ops (N=10)	Ops (N=100)	Operații (N=1000)	Timp (N=1000)
O(1)	1	1	1	1 ns
O(log2(N))	4	7	10	10 ns
O(N)	10	100	1000	1 us
O(Nlog2N)	40	700	10_000	10 us
O(N^2)	100	10_000	1000_000	1 ms
O(N^3)	1000	1000_000	1000_000_000	1 s
O(2^N)	**1024**	10^{31}	10^{302}	>>> universul
O(N!)	3628_800	10^{161}	10^{2567}	I give up