Analiza Algoritmilor

Laboratorul 2 – Complexitatea algoritmilor

Rezolvare

Vlad Dogaru
ddvlad@gmail.com

1 noiembrie 2008

Problema 1

Se dă un vector de numere întregi. Să se parcurgă vectorul până la întâlnirea primului număr negativ şi să se afişeze numerele pare. Nu se garantează că vectorul conţine un număr negativ.

a) Scrieţi un algoritm care rezolvă problema.

b) Calculaţi complexitatea exactă a algoritmului folosind metrica neomogenă (toate operaţiile şi doar operaţiile critice) şi stabiliţi clasa de complexitate.

Rezolvare

Fie v vectorul dat la intrare şi n lungimea acestuia. Considerăm elementele vectorului indexate începând cu 1. Algoritmul este următorul:

Algoritm	**Cost**	**Număr repetări**
$k = 1$	c_1	1
while $(v[k] \geq 0) \wedge (k \leq n)$	c_2	m
if $(v[k]\%2 == 0)$ **then**	c_3	$m - 1$
print $v[k]$	c_4	p
$k = k + 1$	c_5	$m - 1$

În algoritmul de mai sus am folosit următoarele notaţii suplimentare:

- m este poziţia în vector pe care apare primul număr negativ. Este evident că $1 \leq m \leq n+1$. Cu alte cuvinte, nu se garantează că apare cel puţin un număr negativ în vector.

- p este numărul de numere pare care apar înaintea primului număr negativ din vector. Este valabilă relaţia $0 \leq p < m$. Cele două numere nu pot fi egale, pentru că numărul de pe poziţia m este cel negativ care cauzează oprirea algoritmului.

Complexitatea exactă folosind toate operaţiile

Cazul cel mai defavorabil corespunde $m = n+1$ şi $p = m-1 = n$, adică vectorul nu conţine numere negative, iar toate elementele vectorului sunt pare. Pentru acest caz, numărul de operaţii va fi:

$$\begin{aligned}
\mathcal{C}(n) &= c_1 + mc_2 + (m-1)c_3 + pc_4 + (m-1)c_5 \\
&= c_1 + (n+1)c_2 + nc_3 + nc_4 + nc_5 \\
&= (c_2 + c_3 + c_4 + c_5)n + (c_1 + c_2) \\
&= \mathcal{O}(n)
\end{aligned}$$

Cazul cel mai favorabil corespunde unui vector care are pe prima poziţie un număr negativ, astfel că nu se tipăreşte nimic şi bucla `while` nu se execută. Matematic, condiţiile sunt $m = 1$ şi $p = 0$. Numărul de operaţii va fi:

$$\begin{aligned}
\mathcal{C}(n) &= c_1 + mc_2 + (m-1)c_3 + pc_4 + (m-1)c_5 \\
&= c_1 + c_2 + 0 + 0 + 0 \\
&= c_1 + c_2 \\
&= \mathcal{O}(1)
\end{aligned}$$

Cazul mediu corespunde următoarei situaţii (din moment ce nu ni s-a specificat o distribuţie a elementelor):

- primul număr negativ apare la jumătatea vectorului, deci $m = \left[\frac{n}{2}\right]$.

- în prima jumătate a vectorului, jumătate dintre elemente sunt pare: $p = \left[\frac{m}{2}\right] = \left[\frac{n}{4}\right]$.

Atenţie: Am folosit notaţia $[x]$ pentru $\lfloor x \rfloor$ sau $\lceil x \rceil$.

Costul în această situaţie va fi:

$$
\begin{aligned}
\mathcal{C}(n) &= c_1 + mc_2 + (m-1)c_3 + pc_4 + (m-1)c_5 \\
&= c_1 + \frac{n}{2}c_2 + \left(\frac{n}{2} - 1\right)c_3 + \frac{n}{4}c_4 + \left(\frac{n}{2} - 1\right)c_5 \\
&= \left(\frac{c_2}{2} + \frac{c_3}{2} + \frac{c_4}{4} + \frac{c_5}{2}\right)n + \left(c_1 - \frac{c_3}{2} - \frac{c_5}{2}\right) \\
&= \mathcal{O}(n)
\end{aligned}
$$

Complexitatea exactă folosind operaţiile critice

Pentru **cazul cel mai defavorabil**, operaţiile din bucla `while` sunt cele critice. Deci,

$$
\begin{aligned}
\mathcal{C}(n) &= (m-1)c_3 + pc_4 + (m-1)c_5 \\
&= nc_3 + nc_4 + nc_5 \\
&= (c_3 + c_4 + c_5)n \\
&= \mathcal{O}(n)
\end{aligned}
$$

În **cazul cel mai favorabil**, operaţiile din bucla `while` nu se execută niciodată, lăsând iniţializarea şi condiţia buclei drept operaţii critice. Costul va fi:

$$
\begin{aligned}
\mathcal{C}(n) &= c_1 + c_2 \\
&= \mathcal{O}(1)
\end{aligned}
$$

Pentru **cazul mediu**, considerăm drept operaţii critice testul de paritate şi incrementarea contorului `k`. Rezultă:

$$
\begin{aligned}
\mathcal{C}(n) &= (m-1)c_3 + (m-1)c_5 \\
&= \left(\frac{n}{2} - 1\right)c_3 + \left(\frac{n}{2} - 1\right)c_5 \\
&= \left(\frac{c_3}{2} + \frac{c_5}{2}\right)n + \left(\frac{c_3}{2} - \frac{c_5}{2}\right) \\
&= \mathcal{O}(n)
\end{aligned}
$$

Observaţie: am fi putut alege drept operaţie critică şi tipărirea, pentru că ea se repetă, în cazul mediu, de $\left[\frac{n}{4}\right]$ ori, deci numărul de repetări este tot liniar în n.

Problema 2

Să se demonstreze următoarele relaţii:

a) $n^2 - n = \mathcal{O}(n)$

b) $n^2 - n \neq o(n)$

c) $n^3 - n^2 = \Theta(n^3)$

d) $f(n) = \mathcal{O}(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$

e) $f(n) = \Theta(g(n)) \Leftrightarrow \begin{cases} f(n) = \mathcal{O}(g(n)) \\ f(n) = \Omega(g(n)) \end{cases}$

Rezolvare

Pentru a demonstra relaţiile vom utiliza definiţiile şi proprietăţile elementare prezentate la curs.

a) $\mathcal{O}(g(n)) := \left\{f : \mathbb{N}^* \to \mathbb{R}_+^* | \exists c \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^*, f(n) \leq cg(n), \forall n \geq n_0\right\}$ Deci, pentru a demonstra apartenenţa funcţiei $n^2 - n$ la clasa de complexitate $\mathcal{O}(n^2)$, este suficient să găsim o pereche (c, n_0) care verifică relaţia.

$$n^2 - n \leq cn^2, \forall n \geq n_0 \Leftrightarrow$$
$$(c-1)n^2 + n \geq 0, \forall n \geq n_0$$

Fixăm $c = 1$. Relaţia devine

$$n \geq 0, \forall n \geq n_0$$

Deci pentru $c = 1$ şi $n_0 = 1$, relaţia este verificată.

b) Pentru a demonstra relaţia, folosim proprietatea

$$f(n) = o(g(n)) \Rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

Dar

$$\lim_{n\to\infty} \frac{n^2 - n}{n^2} = 1$$

Deci egalitatea iniţială nu este verificată.

Altfel, folosind definiţia:

Trebuie să arătăm că $\forall c \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^*, cn^2 - n \leq n^2, \forall n \geq n_0$.

Fie $c = \frac{1}{2}$. Relaţia de demonstrat devine:

$$\exists n_0 \in \mathbb{N}^*, -\frac{1}{2}n^2 + n \geq 0, \forall n \geq n_0$$

Pentru că n este pozitiv şi nenul, putem împărţi relaţia prin el, păstrând sensul inegalităţii:

$$\exists n_0 \in \mathbb{N}^*, -\frac{1}{2}n + 1 \geq 0, \forall n \geq n_0$$

Relaţia la care am ajuns este falsă, deoarece funcţia liniară din stânga inegalităţii tinde la $-\infty$. În concluzie, am demonstrat că există o constantă c pentru care nu putem găsi un n_0 care satisface relaţia de apartenenţă la mulţimea $o(n^2)$.

c)

$$\begin{aligned}\Theta(g(n)) &= \{f : \mathbb{N}^* \to \mathbb{R}_+^* | \exists c_1, c_2 \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^*, \\ &\quad c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0\}\end{aligned}$$

Inegalitatea dublă se transformă în două inegalităţi care trebuie verificate simultan:

$$\begin{aligned} &\begin{cases} (c_1 - 1)n^3 + n^2 \leq 0 \\ (c_2 - 1)n^3 + n^2 \geq 0 \end{cases}, \forall n \geq n_0 \\ \Leftrightarrow &\begin{cases} (c_1 - 1)n + 1 \leq 0 \\ (c_2 - 1)n + 1 \geq 0 \end{cases}, \forall n \geq n_0 \end{aligned}$$

Alegem c_1 şi c_2 astfel încât funcţiile din stânga semnului de inegalitate să fie prima descrescătoare, iar cea de-a doua crescătoare. Fie $c_1 = 0.5$ şi $c_2 = 2$. Relaţiile devin:

$$\begin{cases} -0.5n + 1 \leq 0 \\ n + 1 \geq 0 \end{cases}, \forall n \geq n_0$$

Pentru $n_0 = 2$, inegalităţile se verifică şi relaţia iniţială este demonstrată.

d)

$$\begin{aligned}
& f(n)=\mathcal{O}(g(n)) \Leftrightarrow \exists c \in \mathbb{R}_{+}^{*}, \exists n_0 \in \mathbb{N}^{*}, f(n) \leq cg(n) | : c \\
\Leftrightarrow \quad & \exists c \in \mathbb{R}_{+}^{*}, \exists n_0 \in \mathbb{N}^{*}, \frac{1}{c} f(n) \leq g(n) \forall n \geq n_0 \\
\Leftrightarrow \quad & g(n)=\Omega(f(n))
\end{aligned}$$

Deci există $c' = \frac{1}{c} \in \mathbb{R}_{+}^{*}$ şi $n'_0 = n_0 \in \mathbb{N}^{*}$ care verifică relaţia pentru apartenenţa la clasa de complexitate Ω.

NB: Împărţirea prin c fără a schimba sensul inegalităţii este posibilă datorită $c > 0$.

NB: Demonstraţia se bazează pe implicaţii duble, pentru că nu face apel decât la matematică elementară.

e) Implicaţia directă:

$$\begin{aligned}
& f(n)=\Theta(g(n)) \\
\Rightarrow \quad & \exists c_1, c_2 \in \mathbb{R}_{+}^{*}, \exists n_0 \in \mathbb{N}^{*}, c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \\
\Rightarrow \quad & \exists c_1, c_2 \in \mathbb{R}_{+}^{*}, \exists n_0 \in \mathbb{N}^{*}, \begin{cases} c_1 g(n) \leq f(n) \\ f(n) \leq c_2 g(n) \end{cases} \forall n \geq n_0 \\
\Rightarrow \quad & \begin{cases} f(n)=\Omega(g(n)) \\ f(n)=\mathcal{O}(g(n)) \end{cases}
\end{aligned}$$

Implicaţia inversă:

$$\begin{aligned}
& \begin{cases} f(n)=\mathcal{O}(g(n)) \\ f(n)=\Omega(g(n)) \end{cases} \\
\Rightarrow \quad & \begin{cases} \exists c_1 \in \mathbb{R}_{+}^{*}, \exists n_{01} \in \mathbb{N}^{*}, f(n) \leq c_1 g(n), \forall n \geq n_{01} \\ \exists c_2 \in \mathbb{R}_{+}^{*}, \exists n_{02} \in \mathbb{N}^{*}, c_2 g(n) \leq f(n), \forall n \geq n_{02} \end{cases} \\
\Rightarrow \quad & \exists c_1, c_2 \in \mathbb{R}_{+}^{*}, \exists n_0 \in \mathbb{N}^{*}, c_2 g(n) \leq f(n) \leq c_1 g(n), \forall n \geq n_0 \\
\Rightarrow \quad & f(n)=\Theta(g(n))
\end{aligned}$$

În demonstraţia de mai sus, am notat $n_0 = \max(n_{01}, n_{02})$.