

# ANALIZA ALGORITMILOR

## TEMATICA

### PROBLEMA 1

#### 2.1. CLASE DE COMPLEXITATE

1. •  $\Theta(n \log n) \cup \Omega(n \log n) \neq O(n \log n)$  - ADEVĂRAT

Fie  $f$  o funcție

$$f(n) = \Theta(n \log n) \Rightarrow \exists c_1, c_2 \in \mathbb{R}_+^*, \exists m_0 \in \mathbb{N}^* \text{ astfel încât}$$

$$c_1 n \log n \leq f(n) \leq c_2 n \log n, \forall n \geq m_0$$

$$f(n) = \Omega(n \log n) \Rightarrow \exists c_3 \in \mathbb{R}_+^*, \exists m_0' \in \mathbb{N}^* \text{ astfel încât}$$

$$f(n) \geq c_3 n \log n, \forall n \geq m_0', c_3 \in \mathbb{R}_+^*$$

$$\text{Dim } f(n) \geq c_3 n \log n, \forall n \geq m_0', c_3 \in \mathbb{R}_+^* \Rightarrow$$

$$\Rightarrow f(n) \geq c_3 n \log n, \forall n \geq m_0', c_3 \in \mathbb{R}_+^*$$

$$(\text{deci } \Omega(n \log n) \subseteq O(n \log n))$$

$$\Rightarrow \text{Putem considera } c_2 = c_3 \text{ și } m_0 = m_0' \Rightarrow$$

$$\Rightarrow reziliuarea că c_1 n \log n \leq f(n) \leq c_3 n \log n,$$

$$\forall n \geq m_0' \Rightarrow f(n) \leq c_3 n \log n, \forall n \geq m_0', c_3 \in \mathbb{R}_+^* \Rightarrow$$

$$\Rightarrow f(n) = O(n \log n)$$

$$\Rightarrow \Theta(n \log n) \cup \Omega(n \log n) \subseteq O(n \log n)$$

Se pune problema dacă  $O(n \log n) \subseteq \Theta(n \log n) \cup \Omega(n \log n)$

Pentru a fi demonstrată dubla inclusiune și deci egalitatea.

Dacă  $O(n \log n) \not\subseteq \Theta(n \log n)$

$$\text{în } \Theta(n \log n) = O(n \log n) \cap \Omega(n \log n)$$

$\Rightarrow$  Dacă  $\exists f$  o funcție, astfel încât  $f \notin \Omega(n \log n)$  și  $f = O(n \log n)$ , atunci  $O(n \log n) \subseteq \Theta(n \log n) \cup \Omega(n \log n)$

O astfel de funcție ar putea fi una care sare la un

moment dat la valori mult mai mari decât în  $O(n \log n)$ . În acestă funcție, afirmația este ADEVĂRATĂ.

$$\bullet \sqrt{n} \log n! = \Theta(n^{\frac{3}{2}} \log n) - \text{ADEVARAT}$$

$$\sqrt{n} \log n! = \Theta(n^{\frac{3}{2}} \log n) \Rightarrow$$

$\Rightarrow \exists c_1, c_2 \in \mathbb{R}_+^*, \exists m_0 \in \mathbb{N}^*$  astfel incat

$$c_1 n^{\frac{3}{2}} \log n \leq \sqrt{n} \log n! \leq c_2 n^{\frac{3}{2}} \log n \Rightarrow$$

$$\Rightarrow c_1 n^{\frac{3}{2}} \log n \leq n^{\frac{1}{2}} \log n! \leq c_2 n^{\frac{3}{2}} \log n / : n^{\frac{1}{2}} \Rightarrow$$

$$\Rightarrow c_1 n \log n \leq \log n! \leq c_2 n \log n$$

$$\Rightarrow \log n^{c_1 n} \leq \log n! \leq \log n^{c_2 n} \quad \left\{ \Rightarrow \text{Baza logaritmilor este } 2 \right.$$

$$\Rightarrow n^{c_1 n} \leq n! \leq n^{c_2 n}$$

Trebuie sa demonstram ca  $\exists c_1, c_2 \in \mathbb{R}_+^* \text{ si } m_0 \in \mathbb{N}^*$  putină core inegalitate să fie adevarată pentru  $\forall n \geq m_0$

Pentru  $c_2 = 1 \Rightarrow n! \leq n^n$ , deci

$$n! = \underbrace{1 \cdot 2 \cdot \dots \cdot n}_{n \text{ elemente}}, \text{ si } n^n = \underbrace{n \cdot n \cdot \dots \cdot n}_{n \text{ elemente}}, \text{ si } n \geq n-1 \geq \dots \geq 1 \Rightarrow$$

$$\Rightarrow n^n \geq n!, \text{ deci } c_2 = 1 \in \mathbb{R}_+^*, n \geq 1$$

Trebuie să găsim  $c_1 \in \mathbb{R}_+^*$  putină core  $n^{c_1 n} \leq n!$

Ne interesează la fel de ce logaritmi

$$\Rightarrow c_1 n \log n \leq \log n! \Rightarrow c_1 \leq \frac{\log n!}{n \log n}$$

$$\text{Calculăm } l = \lim_{n \rightarrow \infty} \frac{\log n!}{n \log n}$$

$$\text{Unde } a_m = \log n!, b_m = m \log n, m \in \mathbb{N}^*$$

(folosesc lema Cesaro - Stolz)

$$b_m \rightarrow \infty \text{ (strict crescător)}$$

$$\Rightarrow vom calcula limita: l' = \lim_{m \rightarrow \infty} \frac{a_{m+1} - a_m}{b_{m+1} - b_m} =$$

$$= \lim_{m \rightarrow \infty} \frac{\log((m+1)!)}{(m+1) \log(m+1) - m \log m} = \lim_{m \rightarrow \infty} \frac{(\log 1 + \log 2 + \dots + \log(m+1)) - (\log 1 + \dots + \log m)}{m \log(m+1) - m \log m + \log(m+1)} =$$

$$= \lim_{n \rightarrow \infty} \frac{\log(n+1)}{n \log\left(\frac{n+1}{n}\right) + \log(n+1)} =$$

$$= \lim_{n \rightarrow \infty} \frac{\log(n+1)}{\log\left(1 + \frac{1}{n}\right)^n + \log(n+1)} \quad \left. \begin{array}{l} \\ \\ \text{dor } \left(1 + \frac{1}{n}\right)^n \rightarrow e \text{ (limite remarcabila)} \\ \Rightarrow \log\left(1 + \frac{1}{n}\right)^n \rightarrow \log e \end{array} \right\} \Rightarrow l' = L \in \overline{\mathbb{R}} \Rightarrow l = L$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{\log n!}{n \log n} = L$$

Deci trebuie să găsim un  $c_1 < L$

$\Rightarrow$  Putem alege un  $c_1 \in (0, 1)$   $\Rightarrow$

$\Rightarrow c_1 = \frac{1}{2} = 0,5$ , care este valabil pentru  $n \geq 2$   
(limitele sunt la  $L$  dincolo)

(pentru  $n=2 \Rightarrow \frac{\log n!}{n \log n} = 0,5$ )

pentru  $n=3 \Rightarrow \frac{\log n!}{n \log n} > 0,5$  și tot așa)

$\Rightarrow \exists c_1, c_2 \in \mathbb{R}_+^* : c_1 = 0,5, c_2 = 1 \wedge, m_0 = 2 \in \mathbb{N}^*$  astfel încât

$$c_1 n^{\frac{3}{2}} \log n \leq \sqrt{n} \log n! \leq c_2 n^{\frac{3}{2}} \log n$$

$$\Rightarrow \sqrt{n} \log n! = \Theta(n^{\frac{3}{2}} \log n)$$

( $m_0$  este 2, deoarece pentru  $c_2$   $n \geq 1$ , și pt.  $c_1$ ,  $n \geq 2$  –  
mai trebuie să alegem  $m_0$ , care să fie adesea  
pentru ambele  $\Rightarrow m_0 = 2$ )

$$\cdot n^3 \log^4 m = O(n^4) - ADEVĀRAT$$

$n^3 \log^4 m = O(n^4) \Leftrightarrow \exists c \in \mathbb{R}_+^*, \exists m_0 \in \mathbb{N}^*$  astfel încât

$$n^3 \log^4 m \leq c \cdot n^4, \forall m \geq m_0$$

$$n^3 \log^4 m \leq c \cdot n^4 / : n^3 =$$

$$\Rightarrow \log^4 m \leq c \cdot m \Rightarrow \frac{\log^4 m}{m} \leq c, \forall m \geq m_0$$

$$\text{fiecum } l = \lim_{n \rightarrow \infty} \frac{\log^4 m}{m} = l$$

$$\left. \begin{array}{l} \log^4(m) \rightarrow \infty \\ n \rightarrow \infty \end{array} \right\} \Rightarrow \text{fiecum l'Hopital}$$

$$\begin{aligned} \Rightarrow l &= \lim_{n \rightarrow \infty} \frac{\log^4 m}{m} \stackrel{l'H}{=} \lim_m \frac{4 \cdot \log^3 m \cdot (\log m)'}{1} = \\ &= \lim_m \frac{4}{\ln 2} \cdot \frac{\log^3 m}{m} \stackrel{l'H}{=} \lim_m \frac{4}{\ln 2} \cdot \frac{3 \cdot \log^2 m}{m \ln^2 2} = \\ &= \lim_m \frac{12}{\ln^2 2} \cdot \frac{\log^2 m}{m} \stackrel{l'H}{=} \lim_m \frac{12}{\ln^2 2} \cdot \frac{2}{m} \cdot \frac{\log m}{m} \stackrel{l'H}{=} \\ &= \lim_m \frac{24}{\ln^3 2} \cdot \frac{1}{m \ln 2} = \lim_m \frac{24}{\ln^3 2} \cdot \frac{1}{m} = 0 \end{aligned}$$

$\Rightarrow$  putem  $m \rightarrow \infty \Rightarrow$  limita este 0.

$$\Rightarrow \exists c \in \mathbb{R}_+^* \text{ astfel încât } \frac{\log^4(m)}{m} \leq c, \forall m \geq m_0$$

Trăznit să găsim  $m_0$  astfel încât core  $\frac{\log^4(m)}{m} \leq c, \forall m \geq m_0$

Studiem particularile pe care funcția este crescătoare și, decrescătoare.

$$\text{Fie } g(x) = \frac{\log^4(x)}{x}$$

$$g'(x) = \frac{4 \log^3(x) \cdot \frac{1}{x \ln 2} \cdot x - \log^4(x)}{x^2} =$$

$$= \frac{4x \log^3(x) - x \log^4(x) \ln 2}{x^3 \ln 2}$$

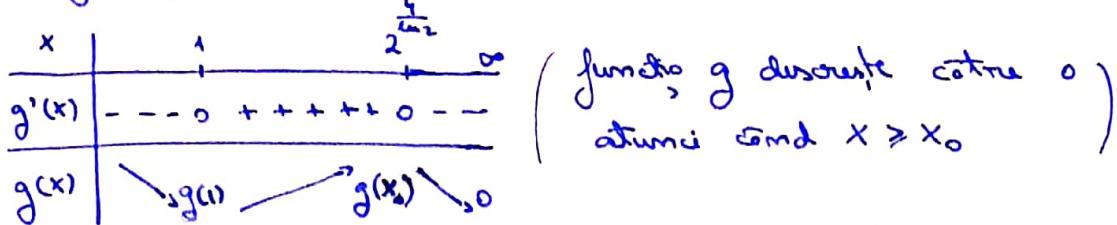
$$g'(x) = 0 \Rightarrow 4x \log^3 x - x \log^4 x \ln 2 = 0 \Rightarrow$$

$$\Rightarrow x = 0 \text{ sau } \log^3 x = 0 \text{ sau } \log x = \frac{4}{\ln 2}$$

$x=0$  nu este lumen pentru că avem la numitor  $x$ .

$$\log^3 x = 0 \Rightarrow x = 1.$$

$$\log x = \frac{4}{\ln 2} \Rightarrow x = 2^{\frac{4}{\ln 2}} = x_0$$



(funcție  $g$  descrescă către 0)  
atunci și  $x \geq x_0$

$$\Rightarrow De la x_0 \text{ înseamnă } \Rightarrow \frac{\log^4 m}{m} \text{ merge să tindă la } 0$$

$$2^{\frac{4}{\ln 2}} < 64 \Rightarrow \text{Atenție } m_0 = 64$$

calculăm c-ul în funcție de  $64 = m_0 \Rightarrow$

$$\Rightarrow \frac{\log^4 m_0}{m_0} = \frac{64^4}{64} = \frac{81}{4} = c$$

$$\Rightarrow \left\{ \begin{array}{l} c = \frac{81}{4} \in \mathbb{R}_+^*, \\ m_0 = 64 \text{ astfel încât} \end{array} \right.$$

$$m^3 \log^4 m \leq c \cdot m^4, \forall m \geq m_0.$$

$$\Rightarrow m^3 \log^4 m = O(m^4)$$

$$\log^{2020} n = g(n) - \text{ADEVARAT}$$

$$\log^{2020} n = g(n) \Leftrightarrow \forall c \in \mathbb{R}_+^*, \exists m_0 \in \mathbb{N}^* \text{ astfel incat}$$

$$\log^{2020} n < c \cdot n$$

$$l = \lim_{n \rightarrow \infty} \frac{\log^{2020} n}{n}$$

$$l \stackrel{L'H}{=} \lim_{n \rightarrow \infty} \frac{2020}{\ln 2} \cdot \frac{\log^{2019} n}{n} \stackrel{L'H}{=} \lim_{n \rightarrow \infty} \frac{2020 \cdot 2019}{\ln^2 2} \cdot \frac{\log^{2019} n}{n} = \\ = \dots = \lim_{n \rightarrow \infty} \frac{2020!}{\ln^{2020} 2} \cdot \frac{1}{n} = 0$$

(Avem aplicat l'Hopital in mod repetat, in mod similar ca in modul in care am rezolvat exercitiul anterior)

$$\lim_{n \rightarrow \infty} \frac{\log^{2020} n}{n} = 0 \Rightarrow \forall c \in \mathbb{R}_+^* \Rightarrow \frac{\log^{2020} n}{n} < c, \forall n \geq n_0$$

$$\Rightarrow \log^{2020} n = g(n)$$

2.

Algoritm 2 ( $A[0 \dots n], m$ )

	Cost	Repetitii
for $i = 1 \dots m$	$c_1$	$m+1$
if ( $A[i] > 0$ )	$c_2$	$m+1$
for $j = 1 \dots i$	$c_3$	$\sum_{i=1}^n (t_i + 1)$
if ( $A[j] \bmod 2 == 0$ )	$c_4$	$\sum_{i=1}^n t_i$
for $k = 1 \dots j$	$c_5$	$\sum_{i=1}^n \sum_{j=1}^{t_i} (t_j + 1)$
$S = S + i + j + k$	$c_6$	$\sum_{i=1}^n \sum_{j=1}^{t_i} t_j$
return $S$ .	$c_7$	1.

Voi folosi metoda numogemă pentru a afla complexitatea algoritmului.

Total operatii:

$$T(n) = c_1(m+1) + c_2 \cdot m + c_3 \sum_{i=1}^n (t_i + 1) + c_4 \sum_{i=1}^n t_i + c_5 \sum_{i=1}^n \sum_{j=1}^{t_i} (t_j + 1) + \\ + c_6 \sum_{i=1}^n \sum_{j=1}^{t_i} t_j + c_7 \cdot 1$$

a) cazul general:

În cazul general nu se întâlnește deocamdată al doilea

for și, în ceea ce e după (în afara de return statement).

Acest lucru se întâmplă atunci când toate elementele din A sunt negative.

$$T(n) = c_1(m+1) + c_2 \cdot m + c_7 \cdot 1 = (c_1 + c_2)m + c_1 + c_7 = \\ = m(c_1 + c_2) + c_1 + c_7 \Rightarrow T(n) = \Theta(n)$$

b) costul de operare este:

costul de operare nu este constant cind toate elementele sunt pozitive, iar toate elementele sunt multiplii de 2

Astfel,  $t_i = i$ , iar  $t_j = j$ .

(Pentru fiecare element  $i$ , se va intra în fiecare for - la fel ca pentru  $j$ )

$$\begin{aligned}
 \Rightarrow T(n) &= c_1 \cdot (m+1) + c_2 \cdot m + c_3 \sum_{i=1}^m (i+1) + c_4 \sum_{i=1}^m i + c_5 \sum_{i=1}^m \sum_{j=1}^{i-1} (j+1) + \\
 &+ c_6 \sum_{i=1}^m \sum_{j=1}^{i-1} j + c_7 \cdot 1 = c_1 \cdot (m+1) + c_2 \cdot m + c_3 \cdot \left( \frac{m(m+1)}{2} + m \right) + \\
 &+ c_4 \cdot \frac{m(m+1)}{2} + c_5 \cdot \sum_{i=1}^m \left( \frac{i(i+1)}{2} + i \right) + c_6 \cdot \sum_{i=1}^m \left( \frac{i(i+1)}{2} \right) + c_7 = \\
 &= c_1 \cdot (m+1) + c_2 \cdot m + c_3 \cdot \frac{m^2}{2} + c_3 \cdot \frac{m}{2} + c_3 \cdot m + c_4 \cdot \frac{m^2}{2} + c_4 \cdot \frac{m}{2} + \\
 &+ c_5 \left( \sum_{i=1}^m \left( \frac{i^2}{2} + \frac{i}{2} + i \right) \right) + c_6 \left( \sum_{i=1}^m \left( \frac{i^2}{2} + \frac{i}{2} \right) \right) + c_7 = \\
 &= \left( c_1 + c_2 + \frac{c_3}{2} + c_3 + \frac{c_4}{2} \right) m + c_3 \frac{m^2}{2} + c_4 \frac{m^2}{2} + c_1 + c_4 + \\
 &+ c_5 \cdot \left( \frac{1}{2} \cdot \frac{m(m+1)(2m+1)}{6} + \frac{1}{2} \cdot \frac{m(m+1)}{2} + \frac{m(m+1)}{2} \right) + \\
 &+ c_6 \left( \frac{1}{2} \cdot \frac{m(m+1)(2m+1)}{6} + \frac{1}{2} \cdot \frac{m(m+1)}{2} \right) = \\
 &= \left( c_1 + c_2 + \frac{c_3}{2} + c_3 + \frac{c_4}{2} \right) m + \left( \frac{c_3}{2} + \frac{c_4}{2} \right) m^2 + c_1 + c_4 + \\
 &+ c_5 \cdot \frac{(m^2+m)(2m+1)}{6} + \frac{c_5}{4} (m^2+m) + \frac{c_5 m^2}{2} + \frac{m}{2} c_5 + \\
 &+ \frac{c_6}{12} (m^2+m)(2m+1) + \frac{c_6}{4} (m^2+m) = \\
 &= \left( c_1 + c_2 + \frac{3}{2} c_3 + \frac{c_4}{2} \right) m + \left( \frac{c_3}{2} + \frac{c_4}{2} \right) m^2 + c_1 + c_4 + \\
 &+ \frac{c_5}{12} (2m^3 + 3m^2 + m) + \frac{c_5}{4} m^2 + \frac{c_5}{2} m^2 + \frac{c_5}{5} m + \frac{m}{2} c_5 + \\
 &+ \frac{c_6}{12} (2m^3 + 3m^2 + m) + \frac{c_6}{4} m^2 + m =
 \end{aligned}$$

$$= \left( \frac{c_3}{6} + \frac{c_6}{6} \right) m^3 + \left( \frac{c_3}{2} + \frac{c_4}{2} + c_5 + \frac{c_6}{2} \right) m^2 + \\ + \left( c_1 + c_2 + \frac{3}{2} c_3 + \frac{c_4}{2} + \frac{5}{6} c_5 + \frac{13}{12} c_6 \right) + c_1 + c_7 \Rightarrow T(m) = \Theta(m^3)$$

c) corul median

În corul median voi considera că jumătate din elementele lui A sunt pozitive, iar jumătate din cele pozitive se repetă cu 2.

$$\Rightarrow t_{ij} = \frac{i}{2}, \text{ și } t_{j\bar{j}} = \frac{j}{2}$$

$$\Rightarrow T(m) = c_1(m+1) + c_2 \cdot m + c_3 \sum_{i=1}^{m-1} \left( \frac{i}{2} + 1 \right) + c_4 \sum_{i=1}^{m-1} \frac{i}{2} + c_5 \sum_{i=1}^{m-1} \sum_{j=1}^{i/2} \left( \frac{j}{2} + 1 \right) + \\ + c_6 \sum_{i=1}^{m-1} \sum_{j=1}^{i/2} \frac{j}{2} + c_7 \cdot 1 = \\ = c_1(m+1) + c_2 \cdot m + c_3 \sum_{i=1}^{m-1} \frac{i}{2} + c_3 \sum_{i=1}^{m-1} 1 + \frac{c_4}{2} \sum_{i=2}^{m-1} i + \\ + c_5 \sum_{i=1}^{m-1} \left( \sum_{j=1}^{i/2} \frac{j}{2} + \sum_{j=1}^{i/2} 1 \right) + c_6 \sum_{i=1}^{m-1} \left( \frac{1}{2} \cdot \frac{i}{2} \cdot \frac{i/2+1}{2} \right) + c_7 \cdot 1 = \\ = c_1(m+1) + c_2 \cdot m + \frac{c_3}{2} \cdot \frac{m(m+1)}{2} + m c_3 + \frac{c_4}{2} \cdot \frac{m(m+1)}{2} + \\ + c_5 \sum_{i=1}^{m-1} \left( \frac{1}{2} \cdot \frac{i}{2} \cdot \frac{i/2+1}{2} + \frac{i}{2} \right) + c_6 \cdot \frac{1}{2} \cdot \frac{1}{2} \sum_{i=1}^{m-1} \left( \frac{i^2}{4} + \frac{i}{2} \right) + c_7 \cdot 1 = \\ = c_1(m+1) + c_2 \cdot m + \frac{c_3}{4} (m^2 + m) + m c_3 + \frac{c_4}{4} (m^2 + m) + \\ + c_5 \frac{1}{4} \cdot \sum_{i=1}^{m-1} \left( \frac{i^2}{4} + \frac{i}{2} \right) + \frac{c_5}{2} \sum_{i=1}^{m-1} i + \frac{c_6}{4} \cdot \frac{1}{4} \cdot \frac{m(m+1)(2m+1)}{6} + \\ + \frac{c_6}{4} \cdot \frac{1}{2} \cdot \frac{m(m+1)}{2} + c_7 = c_1(m+1) + c_2 \cdot m + \frac{c_3}{4} m^2 + \frac{c_3}{4} m + m c_3 + \frac{c_4}{4} m^2 + \frac{c_4}{4} m + \\ + \frac{c_5}{16} \cdot \frac{m(m+1)(2m+1)}{6} + \frac{c_5}{8} \cdot \frac{m(m+1)}{2} + \frac{c_5}{2} \cdot \frac{m(m+1)}{2} + \frac{c_6}{16} \cdot \frac{m(m+1)(2m+1)}{6} + \\ + \frac{c_6}{8} \cdot \frac{m(m+1)}{2} = c_1(m+1) + c_2 \cdot m + \frac{c_3}{4} m^2 + \frac{c_3}{4} m + c_3 m + \frac{c_4}{4} m^2 + \frac{c_4}{4} m + \\ + \frac{c_5}{96} \cdot (2m^3 + 3m^2 + m) + \frac{c_5}{16} (m^2 + m) + \frac{c_5}{4} (m^2 + m) + \frac{c_6}{96} (2m^3 + 3m^2 + m) + \frac{c_6}{16} (m^2 + m) =$$

$$\begin{aligned}
&= c_1(m+1) + c_2 \cdot m + \left( \frac{c_3}{4} + \frac{c_4}{4} + \frac{c_5}{32} + \frac{c_5}{16} + \frac{c_5}{4} + \frac{c_6}{32} + \frac{c_6}{16} \right) m^2 + \frac{c_3}{4} m + c_3 \cdot m \\
&+ \frac{c_4}{4} m + \left( \frac{c_5}{48} + \frac{c_6}{48} \right) m^3 + \frac{c_5}{96} m + \frac{c_5}{16} m + \frac{c_5}{4} m + \frac{c_6}{96} \cdot m + \frac{c_6}{16} m + c_7 = \\
&= \left( \frac{c_5}{48} + \frac{c_6}{48} \right) m^3 + \left( \frac{c_3}{4} + \frac{c_4}{4} + \frac{11c_5}{32} + \frac{3c_6}{32} \right) m^2 + (c_1 + c_2 + \frac{5}{3} c_3 + \frac{c_4}{4} + \right. \\
&\left. + \frac{31}{96} c_5 + \frac{4}{96} c_6 \right) m + c_7 + c_1 \\
\Rightarrow T(n) &= \Theta(n^3)
\end{aligned}$$

Algoritm 2 ( $A[0 \dots m]$ , $m$ )	Cost	Repetiti
$S = 0$	$c_2$	$\perp$
for $i = 1 \dots m$	$c_2$	$m+1$
for $j = 1 \dots m$	$c_2$	$\sum_{i=1}^m (m+1)$
if ( $A[i] > A[j]$ )	$c_4$	$\sum_{i=1}^m i$
for $k = 1 \dots i$	$c_5$	$\sum_{i=1}^m \sum_{k=1}^i (i+1)$
$S = S + A[i] + A[k]$	$c_6$	$\sum_{i=1}^m \sum_{k=1}^i i$
return $S$	$c_7$	$\perp$

Vă rezolvă folosind metrică neomogenă (toate operațiile + operații urtice).

Cel de-al doilea for este și un  $c_2$  și va executa de  $\sum_{i=1}^m (m+1)$  ori.

Numeșteți de repetitii, al celui de-al treilea for depinde de numărul de rezultate de trase din if. Acest număr este  $t_i$ , patru în ambele cazuri. Cel de-al treilea for se repete de  $i$  ori.

Toate operatiile:

$$T(n) = c_1 + c_2(n+1) + c_3 \cdot \sum_{i=1}^n (n+i) + c_4 \sum_{i=1}^n i + \\ + c_5 \sum_{i=1}^n \sum_{k=1}^{t_i} (i+k) + c_6 \sum_{i=1}^n \sum_{k=1}^{t_i} i + c_7$$

a) cost generalul

$\hat{t}_i$  costul generalul, if-ul va fi intotdeauna fals.  $\hat{t}_i$  astfel ca nu se va intra deloc in ciklul de-al treilea for.

$$\Rightarrow t_i \approx 0$$

$$\Rightarrow T(n) = c_1 + c_2(n+1) + c_3 \cdot n(n+1) + c_4 \cdot n^2 + c_7 =$$

$$= n^2(c_3 + c_4) + n(c_2 + c_3) + c_1 + c_2 + c_7$$

$$\Rightarrow T(n) = \Theta(n^2)$$

b) costul defavorabil

$\hat{t}_i$  costul defavorabil, if-ul va fi true pentru toate j-urile din ciklul de-al doilea for.

$$\Rightarrow t_i = n$$

$$\Rightarrow T(n) = c_1 + c_2(n+1) + c_3 n(n+1) + c_4 n^2 + c_5 \sum_{i=1}^n \sum_{k=1}^n (i+k) + \\ + c_6 \sum_{i=1}^n \sum_{k=1}^{t_i} i + c_7 = c_1 + c_2 n + c_2 + c_3 n^2 + c_3 n + c_4 n^2 + \\ + c_5 \sum_{i=1}^n ((i+1) \cdot n) + c_6 \sum_{i=1}^n n i + c_7 = c_1 + c_2(n+1) + c_3 n(n+1) + \\ + c_4 n^2 + n c_5 \cdot \left( \frac{n(n+1)}{2} + n \right) + c_6 n \frac{n(n+1)}{2} + c_7 = \\ = c_1 + c_2 n + c_2 + c_3 n^2 + c_3 n + c_4 n^2 + \frac{c_5}{2} n^3 + \frac{c_5}{2} n^2 + c_5 n^2 + \\ + \frac{c_6}{2} n^3 + \frac{c_6}{2} n^2 + c_7 = \\ = \left( \frac{c_5}{2} + \frac{c_6}{2} \right) n^3 + (c_3 + c_4 + \frac{3c_5}{2} + \frac{c_6}{2}) n^2 + (c_2 + c_3) n + c_1 + c_2$$

$$\Rightarrow T(n) = \Theta(n^3)$$

c) costul mediu:

$\hat{T}(n)$  costul mediu, doar jumătate din condiții din ilg nu și true

$$\Rightarrow t_i = \frac{n}{2}$$

$$\begin{aligned} \Rightarrow T(n) &= c_1 + c_2(n+1) + c_3 \sum_{i=1}^m (n+i) + c_4 \sum_{i=1}^m i + c_5 \sum_{i=1}^m \sum_{k=1}^{m/2} (i+k) + \\ &+ c_6 \sum_{i=1}^m \sum_{k=1}^{m/2} i + c_7 = c_1 + c_2(n+1) + c_3(m+1)n + c_4m^2 + c_5 \sum_{i=1}^m ((i+1)\frac{m}{2}) + \\ &+ c_6 \sum_{i=1}^m (i\frac{m}{2}) + c_7 = c_1 + c_2(n+1) + c_3m^2 + m + c_4m^2 + \frac{c_5}{2}m \cdot \left( \frac{m(m+1)}{2} + m \right) + \\ &+ c_6 \cdot \frac{m(m+1)}{2} \cdot \frac{m}{2} + c_7 = c_1 + c_2m + c_2 + c_3m^2 + c_3m + c_4m^2 + \frac{c_5}{4}(m^3 + m^2) + \\ &+ \frac{c_5}{2}m^2 + \frac{c_6}{4}(m^3 + m^2) + c_7 = \\ &= \left( \frac{c_5}{4} + \frac{c_6}{4} \right)m^3 + (c_3 + c_4 + \frac{3c_5}{4} + \frac{c_6}{4})m^2 + (c_2 + c_3)m + c_1 + c_2 + c_7 \\ \Rightarrow T(n) &= \Theta(n^3) \end{aligned}$$

Operatii critice:

Operatiile critice sunt operatiile executate mai frecvent sau cele mai complicate din punct de vedere al complexitatii:

a) Cost favorabil  $\rightarrow \hat{T}(n)$  costul favorabil, Operatiile critice sunt  $c_3$  și  $c_4$  (ale careloriile  $c_3$  și  $c_4$ )

$$\begin{aligned} \Rightarrow T(n) &= c_3 \sum_{i=1}^m (n+i) + c_4 \sum_{i=1}^m i = \\ &= c_3 m(n+1) + c_4 m^2 = (c_3 + c_4)m^2 + c_3m \\ \Rightarrow T(n) &= \Theta(m^2) \end{aligned}$$

### b) casul defavorabil

În cazul defavorabil, operării vor fi numai cu  $c_5, c_6$ . (că nu există  $c_5, c_6$ , iar  $t_i = m$ )

$$\begin{aligned}
 \Rightarrow T(n) &= c_5 \sum_{i=1}^m \sum_{k=1}^m (i+k) + c_6 \sum_{i=1}^m \sum_{k=1}^m i = \\
 &= c_5 \sum_{i=1}^m m(i+1) + c_6 \sum_{i=1}^m i \cdot m = m c_5 \left( \frac{m(m+1)}{2} + m \right) + \\
 &\quad + m c_6 \frac{m(m+1)}{2} = c_5 \frac{m}{2} \cdot (m^2 + m^2) + m^2 c_5 + \frac{c_6}{2} (m^3 + m^2) = \\
 &= \left( \frac{c_5}{2} + \frac{c_6}{2} \right) m^3 + \left( \frac{3c_5}{2} + \frac{c_6}{2} \right) m^2 \\
 \Rightarrow T(n) &= \Theta(m^3)
 \end{aligned}$$

### c) cazul mediu.

În cazul mediu, operații vor fi numai tot  $c_5, c_6$ ,  $t_i = \frac{m}{2}$   
 (deoarece jumătate din cărora din if vor fi true).

$$\begin{aligned}
 \Rightarrow T(n) &= c_5 \sum_{i=1}^m \sum_{k=1}^{m/2} (i+k) + c_6 \sum_{i=1}^m \sum_{k=1}^{m/2} i = \\
 &= c_5 \sum_{i=1}^m (i+1) \frac{m}{2} + c_6 \sum_{i=1}^m i \cdot \frac{m}{2} = m \frac{c_5}{2} \left( \frac{m(m+1)}{2} + m \right) + m \cdot c_6 \frac{m(m+1)}{2} = \\
 &= \frac{c_5}{4} (m^3 + m^2) + c_5 \frac{m^2}{2} + \frac{c_6}{2} (m^3 + m^2) = \\
 &= \left( \frac{c_5}{4} + \frac{c_6}{2} \right) m^3 + \left( \frac{3c_5}{4} + \frac{c_6}{2} \right) m^2 \\
 \Rightarrow T(n) &= \Theta(m^3)
 \end{aligned}$$

PROBLEMA 2.

1. 2.

3.

$$\cdot T(m) = \begin{cases} 3T\left(\frac{m-2}{2}\right) + k_2 m^2 & , m \geq 2, k_2 \in \mathbb{R}_+^* \\ k_1 & , m=1, k_1 \in \mathbb{R}_+^* \end{cases}$$

$$T(m) = 3T\left(\frac{m-2}{2}\right) + \Theta(m^2), T(1) = \Theta(1)$$

$$\text{Notation } m = 4m - 2$$

$$\Rightarrow T(4m-2) = 3T\left(\frac{4m-2-2}{2}\right) + \Theta((4m-2)^2)$$

$$T(4m-2) = 3T\left(4 \cdot \frac{m}{2} - 2\right) + \Theta(16m^2 - 16m + 4)$$

$$\text{Wende } \Theta(16m^2 - 16m + 4) = \Theta(m^2)$$

$$\Rightarrow T(4m-2) = 3T\left(4 \cdot \frac{m}{2} - 2\right) + \Theta(m^2)$$

$$\text{Notation } S(m) = T(4m-2), S(1) = \Theta(1)$$

$$\Rightarrow S(m) = 3S\left(\frac{m}{2}\right) + \Theta(m^2)$$

$$S(m) = 3S\left(m/2\right) + \Theta(m^2) | \cdot 3^0$$

$$S(m/2^1) = 3S\left(m/2^1\right) + \Theta\left((m/2^1)^2\right) | \cdot 3^1$$

.....

$$S(m/2^k) = 3S\left(m/2^{k+1}\right) + \Theta\left((m/2^{k+1})^2\right) | \cdot 3^k$$

$$\text{Wende } \frac{m}{2^{k+1}} = 1 \Rightarrow m = 2^{k+1} \Rightarrow k+1 = \lg m \Rightarrow k = \lg m - 1$$

$$\Rightarrow S(m) = 3^{k+1} \Theta(1) + \sum_{i=0}^{k-1} 3^i \cdot \frac{1}{4^{2i}} \cdot m^2 \cdot \Theta(1)$$

$$S(m) = 3^{\lg m} \Theta(1) + \Theta(m^2) \cdot \frac{\left(\frac{3}{16}\right)^{k+1} - 1}{\frac{3}{16} - 1} =$$

$$= m^{\lg 3} \Theta(1) + \Theta(m^2) \cdot \frac{\left(\frac{3}{16}\right)^{\lg m} - 1}{-\frac{13}{16}} =$$

$$\begin{aligned}
 &= \Theta(m \lg^3) + \Theta(m^2) \cdot \left(-\frac{16}{13}\right) \left( \left(\frac{3}{16}\right)^{\lg m} - 1 \right) = \\
 &= \Theta(m \lg^3) + \Theta(m^2) \cdot \left(-\frac{16}{13}\right) \left( m^{\lg \frac{3}{16}} - 1 \right) = \\
 &= \Theta(m \lg^3) + \Theta(m^2) \left( \frac{16}{13} - m^{\lg \frac{3}{16}} \right) \quad \left. \right\} = \\
 &\lg \frac{3}{16} < 0 \\
 &\Rightarrow m^2 + \lg \frac{3}{16} < m^2 \\
 &\Rightarrow \Theta(m^2) \left( \frac{16}{13} - m^{\lg \frac{3}{16}} \right) = \Theta(m^2)
 \end{aligned}$$

De asemenea  $m^{\lg 3} < m^2$  pt. că  $\lg 3 < 2$

$$\Rightarrow S(m) = \Theta(m^2)$$

$$T(4m-2) = S(m) = T(4m-2) = \Theta(m^2)$$

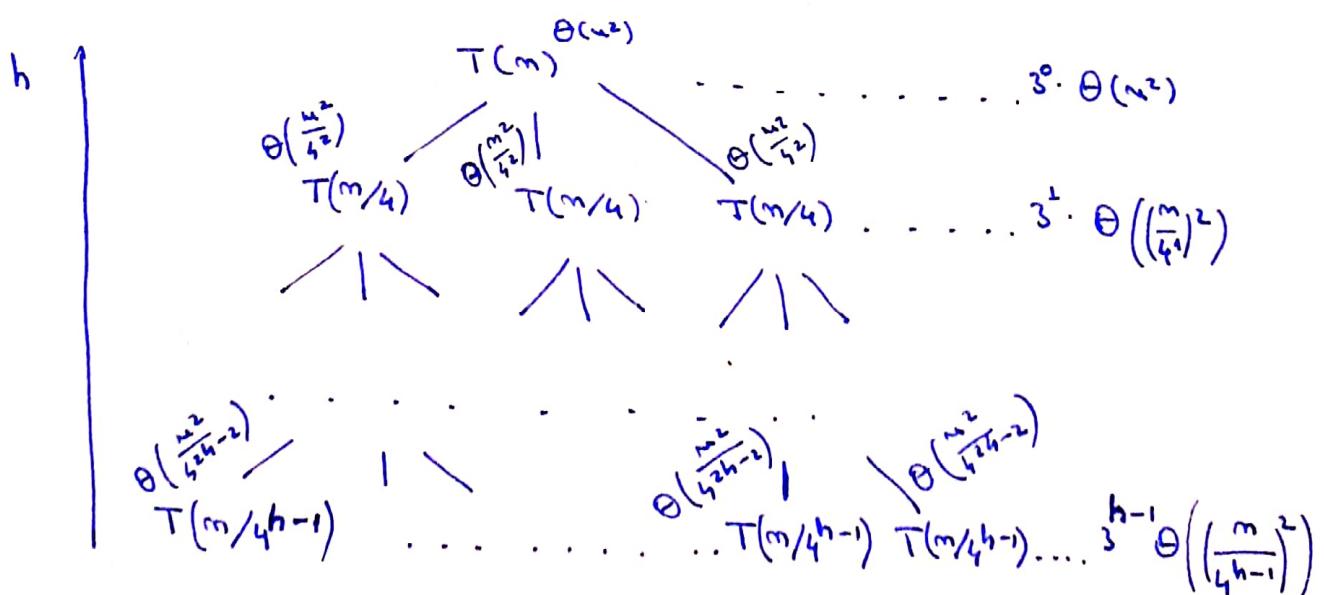
$$m = \frac{m+2}{4}$$

$$\Rightarrow T(m) = \Theta(m^2)$$

$$T(m) = \begin{cases} 3T\left(\frac{m}{4}\right) + k_2 m^2, & m > 1, k_2 \in \mathbb{R}_+^* \\ k_1, & m = 1, k_1 \in \mathbb{R}_+^* \end{cases}$$

$$T(m) = 3T(m/4) + \Theta(m^2), T(1) = \Theta(1)$$

Felurc metoda arboreului de recurență



$h$  - înălțimea, iar  $\frac{m}{4^{h-1}} = 1 \Rightarrow h = \log_4 m + 1$

$$\begin{aligned}
\Rightarrow T(n) &= \sum_{i=0}^{h-1} \left( 3^i \cdot \Theta\left(\frac{n^2}{4^{2i}}\right) \right) = \\
&= \sum_{i=0}^{h-1} \left( \Theta(n^2) \cdot \frac{3^i}{4^{2i}} \right) = \Theta(n^2) \cdot \sum_{i=0}^{h-1} \left( \frac{3}{16} \right)^i = \Theta(n^2) \cdot \sum_{i=0}^{h-1} \left( \frac{3}{16} \right)^i = \\
&= \Theta(n^2) \cdot \frac{\left( \frac{3}{16} \right)^h - 1}{\frac{3}{16} - 1} = \Theta(n^2) \cdot \left( -\frac{16}{13} \right) \left( \left( \frac{3}{16} \right)^{\log_4 n + 1} - 1 \right) = \\
&= \Theta(n^2) \cdot \left( 1 \cdot \frac{16}{13} - \frac{16}{13} \cdot \frac{3}{16} \cdot \left( \frac{3}{16} \right)^{\log_4 n} \right) = \\
&= \Theta(n^2) \cdot \left( \frac{16}{13} - \frac{3}{13} \cdot n^{\log_4 \frac{3}{16}} \right) \\
&\text{, da } \log_4 \frac{3}{16} < 0 \Rightarrow n^2 > n^{2 + \log_4 \frac{3}{16}} \quad \left. \right\} = \\
\Rightarrow T(n) &= \Theta(n^2) \left( \frac{16}{13} - \frac{3}{13} \cdot n^{\log_4 \frac{3}{16}} \right) = \Theta(n^2) \\
\Rightarrow T(n) &= \Theta(n^2)
\end{aligned}$$

$$T(n) = \begin{cases} 3T\left(\frac{n}{4}\right) + n \log n k_2, & n > L, k_2 \in \mathbb{R}_+^* \\ k_1, & n = 1, k_1 \in \mathbb{R}_+^* \end{cases}$$

$$T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n \log n), T(1) = \Theta(1)$$

Rozolvam metoda MASTER.

În cazul 2:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Dacă  $f(n) = \Theta(n^{\log_b a})$  atunci  $T(n) = \Theta(n^{\log_b a} \cdot \lg n)$   
unde  $\lg$  - logarithm în baza 2 din  $n$ .

$$\text{Dacă } a = 3, b = 4 \\
\Rightarrow n^{\log_b a} = n^{\log_4 3}$$

$$\left. \begin{array}{l} f(n) = k_2 n \log n = \Theta(n \log n) \\ \log_4 3 < 1 \end{array} \right\} \Rightarrow f(n) \neq \Theta(n^{\log_b a}) \Rightarrow$$

$\Rightarrow$  nu se poate aplica corol 2.

Încerc să rezolv folosind corol 1:

$$T(n) = aT(n/a) + f(n)$$

corol: Dacă  $\exists \varepsilon > 0$  astfel încât  $f(n) = O(n^{\log_a a - \varepsilon})$

$$\text{atunci } T(n) = \Theta(n^{\log_a a})$$

$$f(n) = k_2 n \log n$$

$$O(n^{\log_a a - \varepsilon}) = O(n^{\log_4 3 - \varepsilon})$$

$$\text{Încerc să demonstreze că } f(n) = O(n^{\log_4 3 - \varepsilon})$$

$\Rightarrow \exists c \in \mathbb{R}_+^*, \exists m_0 \in \mathbb{N}^*$  astfel încât

$$c n^{\log_4 3 - \varepsilon} \geq k_2 n \log n$$

$$\text{, deoarece } \log_4 3 < 1 \mid -\varepsilon \Rightarrow \log_4 3 - \varepsilon < 1 - \varepsilon < 1 \Rightarrow$$

$\Rightarrow n \log n$  crește mai rapid decât  $n^{\log_4 3 - \varepsilon}$ , unde  $\varepsilon > 0$

$\Rightarrow \exists \varepsilon > 0$  astfel încât  $f(n) = O(n^{\log_4 3 - \varepsilon})$

$\Rightarrow$  nu putem aplica corol 1.

Încerc să rezolv folosind corol 3:

$$T(n) = aT(n/a) + f(n)$$

corol 3: Dacă  $\exists \varepsilon > 0$  astfel încât  $f(n) = \omega(n^{\log_a a + \varepsilon})$  și

$\exists c \in (0, 1), m_0 \in \mathbb{N}^*$  astfel încât

$$a f(n/a) \leq c \cdot f(n), \forall n \geq m_0$$

$$\text{atunci } T(n) = \Theta(f(n))$$

Am 2 etape în demonstrație:

Prima etapă arată că  $\exists \varepsilon$ .

$\exists \varepsilon > 0$  astfel încât  $f(n) = \omega(n^{\log_a a + \varepsilon})$

$\Rightarrow \exists c_1 \in \mathbb{R}_+^*, \exists m'_0 \in \mathbb{N}^*$  astfel încât

$$c_1 n^{\log_4 3 + \varepsilon} \leq f(n), \forall n \geq n_0$$

$$\Rightarrow c_1 n^{\log_4 3 + \varepsilon} \leq k_2 n \log n \quad (\because n, \forall n \geq n_0 \Rightarrow)$$

$$\Rightarrow c_1 n^{\log_4 3 + \varepsilon - 1} \leq k_2 \log n$$

$\Rightarrow$  Alegem un  $\varepsilon > 0$  putem avea  $\{c\}$  si  $n_0$  putem avea

$$f(n) = \lceil 2 (n^{\log_4 3 + \varepsilon}) \rceil$$

$\Rightarrow$  Putem alege astfel astfel  $\log_4 3 + \varepsilon - 1 = 0 \Rightarrow \varepsilon = 1 - \log_4 3 > 0$

$$\Rightarrow c_1 n^{\log_4 3 + \varepsilon - 1} = c_1 n^0 = c_1$$

$\Rightarrow k_2 \log n \geq c_1$  unde  $n_0' = 2 \in \mathbb{N}^*$ , iar  $c_1 = k_2 \in \mathbb{R}_+^*$

$$\Rightarrow \varepsilon = 1 - \log_4 3 \quad (1)$$

Apoi avem ca  $\{c \in (0,1)\}, \{n_0 \in \mathbb{N}^*\}$  astfel incat

$$a f(n/k) \leq c \cdot f(n), \forall n \geq n_0$$

$$\Rightarrow 3 \frac{n}{4} \lg \frac{n}{4} \leq c \cdot n \lg n, \forall n \geq n_0$$

$$\stackrel{(1)}{\Rightarrow} \frac{3}{4} \lg \frac{n}{4} \leq c \lg n$$

$$\frac{3}{4} (\lg n - \lg 4) \leq c \lg n, \forall n \geq n_0$$

$$\Rightarrow \frac{3}{4} \cdot \frac{\lg n - \lg 2}{\lg n} = c$$

$$\lg n - \lg 2 \leq \lg n \Rightarrow \frac{\lg n - \lg 2}{\lg n} \leq 1.$$

$$\Rightarrow c = \frac{3}{4} \in (0,1)$$

iar  $n_0 = 2 \in \mathbb{N}^*$ , deoarece  $n \neq 1$  dim moment u creste

logarithm la numitor (logaritmul este mai mare)

$\Rightarrow \{c = \frac{3}{4} \in (0,1), \{n_0 \in \mathbb{N}^*\} \text{ unde } n_0 = 2 \text{ astfel incat}$

$$a f(n/k) \leq c \cdot f(n), \forall n \geq n_0 \quad (2)$$

Din (1) si (2)  $\Rightarrow$  putem scrie următoarele  $\exists =$

$$\Rightarrow T(n) = \Theta(n \log n)$$

$$\cdot T(n) = \begin{cases} 3T\left(\frac{n}{3}\right) + \frac{n}{\log n} k_2, & n > 1, k_2 \in \mathbb{R}_+ \\ k_1, & n = 1, \end{cases}$$

$$T(n) = 3T(n/3) + \Theta(n/\log n), \quad T(1) = \Theta(1)$$

Fiec cu metoda iterative

$$T(n) = 3T(n/3) + \Theta(n/\log n) \quad | \cdot 3^0$$

$$T(n/3^1) = 3T(n/3^2) + \Theta\left(\frac{n}{3}/\log\frac{n}{3}\right) | \cdot 3^1$$

.....

$$T(n/3^k) = 3T(n/3^{k+1}) + \Theta\left(\frac{n}{3^k}/\log\frac{n}{3^k}\right) | \cdot 3^k$$

$$\text{unde } \frac{n}{3^{k+1}} = 1 \Rightarrow n = 3^{k+1} \Rightarrow k = \log_3 n - 1$$

$$T(n) = 3^{k+1} \Theta(1) + \sum_{i=0}^k 3^i \cdot \frac{n}{3^i \log \frac{n}{3^i}} \Theta(1) =$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{\log \frac{n}{3^0}} + \frac{1}{\log \frac{n}{3^1}} + \dots + \frac{1}{\log \frac{n}{3^k}} \right) =$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{\log \frac{n}{3^{\log_3 n - \log_3 1}}} + \dots + \frac{1}{\log \frac{n}{3^{\log_3 n - 2}}} + \frac{1}{\log \frac{n}{3^{\log_3 n - 1}}} \right) =$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{\log \frac{3n}{n}} + \frac{1}{\log \frac{3^2 n}{n}} + \dots + \frac{1}{\log \frac{3^{\log_3 n} n}{n}} \right) =$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{\log 3} + \frac{1}{\log 3^2} + \dots + \frac{1}{\log n} \right) =$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{\log 3} + \frac{1}{2 \log 3} + \dots + \frac{1}{\log 3^n \cdot \log 3} \right) =$$

$$= \Theta(n) + \Theta(n) \cdot \frac{1}{\log 3} \left( \frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{\log 3^n} \right)$$

$$= \Theta(n) + \Theta(n) \left( \frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{\log_3 n} \right)$$

De la analiză stiu că

$$\lim_{n \rightarrow \infty} \frac{1 + \frac{1}{2} + \dots + \frac{1}{n}}{\ln n} = 1 \quad (\text{demonstrat în următorul desfășurare})$$

$$\Rightarrow 1 + \frac{1}{2} + \dots + \frac{1}{n} \sim \ln n$$

pentru numere mari să să consider

$$\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{\log_3 n} \sim \ln(\log_3 n) \sim \log(\log n)$$

$$\Rightarrow T(n) = \Theta(n) + \Theta(n) \log(\log n)$$

Consider  $\ln n \sim \log n$

$$\Rightarrow T(n) = \Theta(n) + \Theta(n) \log(\log n)$$

$$\Rightarrow T(n) = \Theta(n \log(\log n))$$

Jn. Recurvență	Metoda rezervită	Complexitatea
Prima recurvență	Metoda iterativă	$\Theta(n^2)$
A doua recurvență	Metoda arboreului de recurență	$\Theta(n^2)$
A treia recurvență	Metoda Master	$\Theta(n \log n)$
A patra recurvență	Metoda iterativă	$\Theta(n \log(\log n))$

4.

$$\bullet T(n) = \begin{cases} 10T\left(\frac{n}{3}\right) + K_2 n^3, & n > 1, K_2 \in \mathbb{R}_+^* \\ K_1, & n = 1, K_1 \in \mathbb{R}_+^* \end{cases}$$

Din calculul făcut cu metoda iterativă a rezultat

$$T(n) = \Theta(n^3)$$

$$\Rightarrow \text{Luăm } T(n) = \Theta(n^3) \Leftrightarrow$$

$$\Leftrightarrow \exists c_1, c_2 \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^* \text{ astfel încât}$$

$$c_1 n^3 \leq T(n) \leq c_2 n^3, \forall n \geq n_0$$

Care de loc este:

$$n=1: c_1 \leq K_1 \leq c_2$$

$$\Rightarrow n_0 = 1 - \text{care de loc este}$$

Pas de inducție:  $n/3 \rightarrow n$

Hipoteza de inducție

$$c_1 \left(\frac{n}{3}\right)^3 \leq T\left(\frac{n}{3}\right) \leq c_2 \left(\frac{n}{3}\right)^3$$

Astăzi că

$$c_1 n^3 \leq T(n) \leq c_2 n^3$$

$$c_1 \left(\frac{n}{3}\right)^3 \leq T\left(\frac{n}{3}\right) \leq c_2 \left(\frac{n}{3}\right)^3 | \cdot 10$$

$$\Rightarrow 10c_1 \left(\frac{n}{3}\right)^3 \leq 10T\left(\frac{n}{3}\right) \leq 10c_2 \left(\frac{n}{3}\right)^3 | + K_2 n^3 \Rightarrow$$

$$\Rightarrow 10c_1 \left(\frac{n}{3}\right)^3 + K_2 n^3 \leq 10T\left(\frac{n}{3}\right) + K_2 n^3 \leq 10c_2 \left(\frac{n}{3}\right)^3 + K_2 n^3$$

Este îndeajuns să demonstreăm că

$$c_1 n^3 \leq 10 c_1 \left(\frac{n}{3}\right)^3 + K_2 n^3 \text{ și}$$

$$10 c_2 \left(\frac{n}{3}\right)^3 + K_2 n^3 \leq c_2 n^3$$

$$c_1 n^3 \leq 10 c_1 \left(\frac{n}{3}\right)^3 + K_2 n^3 \Rightarrow c_1 \leq 10 \cdot \frac{1}{27} c_1 + K_2 \Rightarrow$$

$$\Rightarrow \frac{17}{27} c_1 \leq K_2 \Rightarrow c_1 \leq \frac{27}{17} K_2 \quad (1)$$

$$10c_2 \left(\frac{n}{3}\right)^3 + K_2 n^3 \leq c_2 n^3 \quad | :n^3$$

$$\frac{10}{27} c_2 + K_2 \leq c_2 \Rightarrow K_2 \leq \frac{17}{27} c_2 \Rightarrow \frac{27}{17} K_2 \leq c_2 \quad (2)$$

Dim corel de baza arem:

$$c_1 \leq K_1 \leq c_2$$

Dim parul de inducție arem (1) și (2):  $c_1 \leq \frac{27}{17} K_2 \leq c_2$

Trebuie să satisfacă condițiile de mai sus pentru ca inducția să fie aderentă:

$$\Rightarrow \begin{cases} c_1 = \min(K_1, \frac{27}{17} K_2) \end{cases}$$

$$\begin{cases} c_2 = \max(K_1, \frac{27}{17} K_2), \end{cases}$$

$$\begin{cases} m_0 = 1 \in \mathbb{N}_0^* \text{ (core de baza),} \end{cases}$$

atfel înțeț

$$T(n) = \Theta(n^3)$$

$$\cdot T(n) = \begin{cases} 7T\left(\frac{n}{3}\right) + K_2 n^2 & , \text{dacă } n > 1, K_2 \in \mathbb{R}_+^* \\ k_1 & , \text{dacă } n = 1, k_1 \in \mathbb{R}_+^* \end{cases}$$

Din calculul făcut cu metoda iterativă a rezultat

$$T(n) = \Theta(n^2)$$

$$\Rightarrow \lim_{n \rightarrow \infty} T(n) = \Theta(n^2) \Leftrightarrow$$

$\Leftrightarrow \exists c_1, c_2 \in \mathbb{R}_+^*, \exists n_0 \in \mathbb{N}^*$  astfel încât

$$c_1 n^2 \leq T(n) \leq c_2 n^2, \forall n \geq n_0$$

Care de ledeză:

$$n=1: c_1 \leq k_1 \leq c_2$$

$$\Rightarrow n_0 = 1 - \text{care de ledeză}$$

Pas de inducție:  $\frac{m}{3} \rightarrow m$

Hipoteza de inducție:

$$c_1 \left(\frac{n}{3}\right)^2 \leq T\left(\frac{n}{3}\right) \leq c_2 \left(\frac{n}{3}\right)^2$$

Astăzi că:

$$c_1 n^2 \leq T(n) \leq c_2 n^2$$

$$c_1 \left(\frac{m}{3}\right)^2 \leq T\left(\frac{m}{3}\right) \leq c_2 \left(\frac{m}{3}\right)^2 | \cdot 4 \Rightarrow$$

$$\Rightarrow 4c_1 \left(\frac{m}{3}\right)^2 \leq 4T\left(\frac{m}{3}\right) \leq 4c_2 \left(\frac{m}{3}\right)^2 | + k_2 n^2 \Rightarrow$$

$$\Rightarrow 4c_1 \left(\frac{m}{3}\right)^2 + k_2 n^2 \leq 4T\left(\frac{m}{3}\right) + k_2 n^2 \leq 4c_2 \left(\frac{m}{3}\right)^2 + k_2 n^2 \Rightarrow$$

$$\Rightarrow 4c_1 \left(\frac{m}{3}\right)^2 + k_2 n^2 \leq T(n) \leq 4c_2 \left(\frac{m}{3}\right)^2 + k_2 n^2$$

Este îndeajuns să demonstreăm că:

$$c_1 n^2 \leq 4c_1 \left(\frac{m}{3}\right)^2 + k_2 n^2 \text{ și}$$

$$4c_2 \left(\frac{m}{3}\right)^2 + k_2 n^2 \leq c_2 n^2$$

$$c_1 m^2 \leq 7 c_1 \left(\frac{n}{3}\right)^2 + K_2 m^2 \quad | :m^2 \Rightarrow$$

$$\Rightarrow c_1 \leq 7 c_1 \cdot \frac{1}{9} + K_2 \Rightarrow \frac{2}{3} c_1 \leq K_2 \quad (1)$$

$$7 c_2 \left(\frac{m}{3}\right)^2 + K_2 m^2 \leq c_2 n^2 \quad | :n^2 \Rightarrow$$

$$\Rightarrow 7 c_2 \cdot \frac{1}{9} + K_2 \leq c_2 \Rightarrow K_2 \leq \frac{2}{3} c_2 \quad (2)$$

Dim corel de lege anum

$$c_1 \leq K_1 \leq c_2$$

Dim paral de inducție anum (1) și (2):  $\frac{2}{3} c_1 \leq K_2 \leq \frac{2}{3} c_2 \Rightarrow$

$$\Rightarrow c_1 \leq \frac{3}{2} K_2 \leq c_2$$

Trebuie să satisfacă condiția de mai sus pentru ca inducție să fie aderată:

$$\Rightarrow \begin{cases} c_1 = \min(K_1, \frac{3}{2} K_2), \\ c_2 = \max(K_1, \frac{3}{2} K_2), \end{cases}$$

$$\begin{cases} m_0 = l \in \mathbb{N}_0^* \text{ (care de lege)} \\ \text{astfel încât} \\ T(n) = \Theta(n^2) \end{cases}$$

1. 3.

### 5. reverse ( $\rightarrow$ )

$$m = \text{size}(\rightarrow)$$

$$\text{if } m <= 1$$

return  $\rightarrow$

else

return concat(reverse(substrings( $\rightarrow$ ,  $m/2$ ,  $m$ )),  
reverse(substrings( $\rightarrow$ , 0,  $m/2$ ))),

$c_1$

$c_2$

$c_3$

Voi considera formula de recurentă:

$$T(n) = \begin{cases} T(L^{n/2}) + T(F_{n/2}) + k_2 n, & k_2 \in \mathbb{R}_+^*, n > 1 \\ k_1, & k_1 \in \mathbb{R}_+^*, n = 1 \end{cases}$$

$$k_1 = c_1 + c_2 - \text{cost de baza}$$

$n k_2 = c_1 + c_3$  - operatia  $c_3$  depinde de apelul functiei substrings (deci nu va fi in timp constant) ( $\frac{n}{2}$  putem primul apel si,  $\frac{n}{2}$  pentru al doilea apel al substrings din return statement). Astfel portea inferioara si portea superioara a  $\frac{n}{2}$  (inferior pt. primul apel de substrings si reverse si superioara pt. urmatorul).

Voi calcula complexitatea folosind metoda iterativa

$$T(n) = 2T(n/2) + \Theta(n), T(1) = \Theta(1)$$

$$T(n) = 2T(n/2) + \Theta(n) | \cdot 2^0$$

$$T(\frac{n}{2}) = 2T(\frac{n}{2^2}) + \Theta(\frac{n}{2}) | \cdot 2^1$$

.....

$$T(\frac{n}{2^k}) = 2T(\frac{n}{2^{k+1}}) + \Theta\left(\frac{n}{2^k}\right) | \cdot 2^k$$

unde  $\frac{n}{2^{k+1}} = 1 \Rightarrow k+1 = \log n \Rightarrow k = \log n - 1$

$$\Rightarrow T(n) = 2^{k+1} \Theta(1) + \sum_{i=0}^k 2^i \cdot \frac{n}{2^i} \cdot \Theta(1) =$$

$$= 2^{\log n} \Theta(1) + \log n \Theta(n) = \Theta(n) + \log n \Theta(n) = \Theta(n \log n).$$

## PROBLEMA 2 - EXPLICATII COMPLEXITATE

La problema 2 am ales să implementez un treap, decice am vrut să combin functionalitățile unei cărora lumea de către în un heap (am folosit min-heap).

Structura folosită arată astăzi:

```
typedef struct treapnode {
```

```
    Type value;
```

```
    int priority;
```

```
    struct treapnode * parent;
```

```
    struct treapnode * left;
```

```
    struct treapnode * right;
```

```
} Node;
```

```
typedef struct sequence {
```

```
    struct treapnode * node;
```

```
    int size;
```

```
} Sequence;
```

Am folosit tipul de date int pt. Type (typedef int Type)  
pentru simplitate și o mai bună organizare, am ales să

folosesc două structuri (treapnode desemnând un nod în treap, iar sequence totă structura). Îndată că folosesc pe post de priorități a fiecărui nod. Astfel, conform proprietăților unei min-heaps, rădăcina arborelui treap va fi cu ce prioritate cea mai mică (indrăznel 0). Pentru a face notabilă mai ușor și a simplifica acest proces de adăugare a unui element și de stergere, am introdus și un pointer spre părintele fiecărui nod (NULL pt. rădăcina). Structura Sequence are în compoziție un nod treapnode (rădăcina - care la rădăcină și se leagă de alte noduri de

tip treapțode) și înțe-ul care reprezintă numărul de elemente din  
treapț.

Fisierele sequence.c conțin mai multe funcții ajutătoare  
pe care ele a căror ajutor a fost dat în enunțul temei.  
Ele sunt folosite pentru a nu modifica antetul și, pentru a sim-  
plifica codul. Fisierele main.c conțin un doar. Acest doar și  
comentarii în sequence.c și, sequence.h pentru a fi de înțeles  
codul. Comanda „make” compilază fisierile. Comanda „make  
run” rulează main-ul. „make clean” sterge executabilele.

Mai jos voi explica complexitățile

- Sequence init(); → O(1)

Funcție returnează structura vidă. (data-structure → size = 0)

Conține un malloc pentru structura Sequence.

De asemenea, nodul rădăcină este initializat cu NULL.

- Sequence insert (Sequence data-structure, Type item, int index)

Funcție de complexitatea O(n)

Este compusă din cinci pași:

1. Este ellocat memorie pentru un nou nod (treapțode) - O(1)

2. Se adaugă nodul după ultimă (fără respectă regulile de  
adăugare într-un arbore binar de cădere) - O(log n)

3. Se fac ajustări necesare pentru a deține elementul treapței  
sprijindu-l în funcție de prioritate (index) - O(log n)

4. Se trage primul nod de la indexul său și se adaugă (astfel încât  
prioritatea acestui element să fie unic) - O(n)

5. Se încrementează size-ul structurii.

Din cui 5 pași (la 2 și 3 fără considerând cazul defavorabil) =,  
⇒ complexitatea este O(n)

- Type lookup (Sequence data-structure, int index)

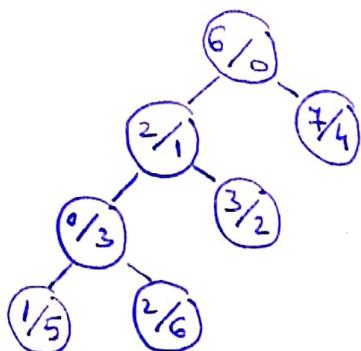
Operații de lookup are în cadrul definiției complexitatea  $O(n)$ .

Functii apelate și funcții ajutătoare (lookup-helper) care returnează referință spre modul căutat.

Functia lookup-helper este recursivă și parcurge tot arborele în ordine postordă cănd găsește elementul cu prioritățea index. Dacă se caute un nod cu un index  $<$  data-structure  $\rightarrow$  size, atunci, dacă în momentul parcurgerii se ajunge la un element cu prioritățea mai mare, atunci nu se mai continuă în jos pe liniile acestui nod.

Așa că, complexitatea nu este în medie  $O(n)$ , dar în majoritatea cazurilor este proporțională cu această complexitate.

Un alt defavorabil ar fi, spre exemplu, acesta:



Dacă se caute elementul de pe poziția 4 (adică 4), atunci se va trage primătoare modulă.

Așa că, pot spune că este  $O(n)$  complexitate.

- Sequence delete (Sequence data-structure, int index)

Această operare are complexitatea  $O(n)$

1. Se caute elementul care trebuie sters  $\rightarrow O(n)$
2. Se stergă nodul  $\rightarrow O(1)$  dacă nodul de către care nodul care lăzi său este frumos (căz frumos) sau dacă succesorul în ordine nu este găsit în  $O(n)$ .

Functia "get-successor" cauță succesorul în ordine. Dacă fil din strângă al lui să fie din dreapta al modului de sters, are un număr de lini împărțit cu  $n$ , atunci complexitatea este  $O(n)$ . Dacă se consideră acest caz, atunci pasul 2 are complexitatea  $O(n)$ .

3. Se face numărarea elementelor în interval mai mare decât al elementului care a fost stors. În acestă situație complexitatea este  $O(n)$

4. Se decrementează size-ul structurii. -  $O(1)$

=> Complexitatea acestei operații este  $O(n)$

- Sequence set (Sequence data-structure, Type item, int index)

Functie calitate în mod similar cu cea de backup elementul al cărei valoare trebuie înlocuit.

Apelează funcție „set-helper” care se realizează în  $O(n)$

=> Complexitatea este  $O(n)$

- int size (Sequence data-structure)

Functie returnează numărul size al structurii Sequence

Aceea complexitatea este  $O(1)$

- Sequence \*split (Sequence data-structure, int index)

Functie este compusă din două pași:

1. Se crează în mod pe rând, fiecare element împărând de la index + 1 în data-structure (se folosește backup) -  $O(n^2)$ .

Tot aici se întâlnește în data-structure și elementul -  $O(n)$

2. Se sterg elementele pur în data-structure și dim

data-structure -  $O(n^2)$

După aceasta se continuă recurența returnat

=> Complexitate:  $O(n^2)$

- Sequence concat (Sequence data-structure1, Sequence data-structure2)

Functie are 2 pași:

1. Recursează în data-structure2 (fiecare element își adună indexul în data-structure1 → size)  $\rightarrow O(m)$  unde m = nr. de elemente din data-structure2.

2. Apelă funcție concat-helper care adaugă trupul 2 în același mod în care se întâlnește un mod într-un binary search tree -  $O(\log n)$

=> Complexitate:  $O(n)$ .