

SD Curs 9

30 Apr 2025

1. Coduri Huffman

Coduri Huffman

Text

a	b	c	d	e	f
45	20	15	13	8	6

frecvența
literelor

Problema

Să se găsească o **codificare** în cod binar pentru fiecare literă astfel încât lungimea totală a textului codificat să fie **minimă**. De asemenea, vrem să putem decodifica textul binar.

$$a = 000$$

$$b = 001$$

$$c = 010$$

$$d = 011$$

$$e = 100$$

$$f = 101$$

→ lungimea totală a textului

$$107 \times 3 = 321$$

Exemplu de codificare a a b d = 000 000 001 011

$$010 101 011 = c f d$$

Notăm alfabetul cu \mathcal{L} și pentru fiecare literă $c \in \mathcal{L}$ notăm $f(c)$ frecvența, dorim să minimizăm

$$\sum_{c \in \mathcal{L}} f(c) \text{ lungimea codului lui } c$$

Exemplu ambiguitate

a	b	c
0	01	10

	a	b	a
?	<u>0</u>	<u>0</u> <u>1</u> <u>0</u>	
	a	a	c

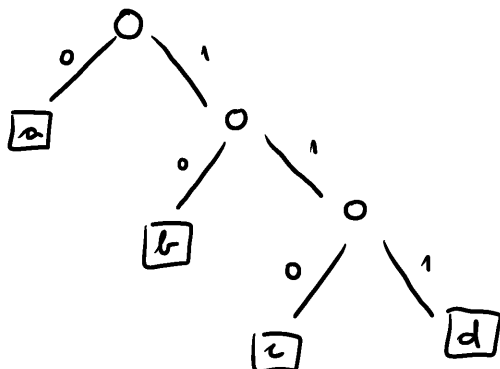
Pentru a putea decodifica, ne folosim pe coduri care nu sunt prefixe ale altor coduri

Exemplu de cod „prefix-free”

a	b	c	d
0	10	110	111

0	10	110	10	111
a	b	c	b	d

Orice cod „prefix-free” poate fi reprezentat ca un arbore binar.



$$\min \sum_{c \in C} f(c) \cdot d(c)$$

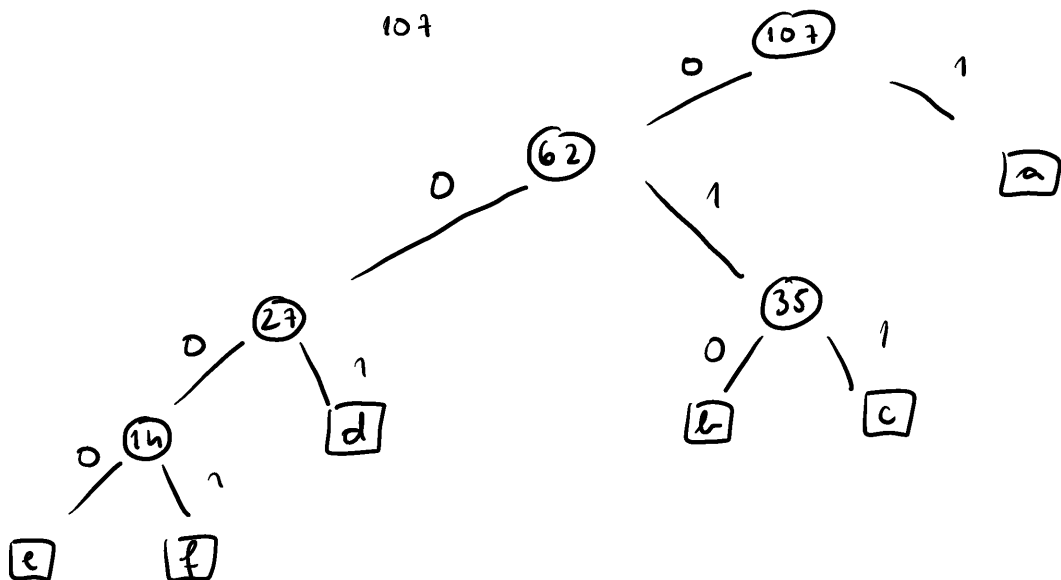
\nearrow
 adancimea funției
 corespunzătoare în
 arborele T

Algoritmul care găsește nodul optim

- luăm 2 litere ca un frecvență minimă. Le punem în arbore ca fi ai unui nou nod care are frecvența suma celor două frecvențe

a	b	c	d	e	f
45	20	15	13	8	6
45	20	15	13	14	
45	20	15	27		
45	35	27			
45		62			

107



a	b	c	d	e	f
1	101	011	001	0000	0001

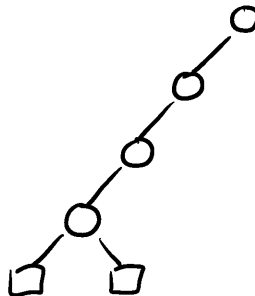
$$45 \cdot 1 + 20 \cdot 3 + 15 \cdot 3 + 13 \cdot 3 + 8 \cdot 4 + 6 \cdot 4 = \boxed{245}$$

vs 321

Obs (exercițiu pt. acasă)

Un cod optim corespunde unui arbore binar plin (în care fiecare nod are excepția frunzelor sau exact 2 fii).

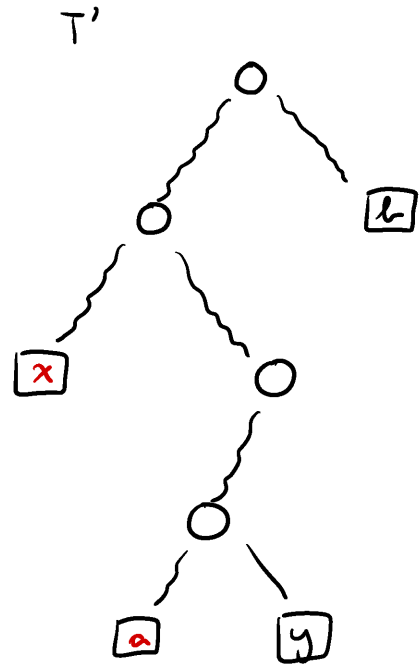
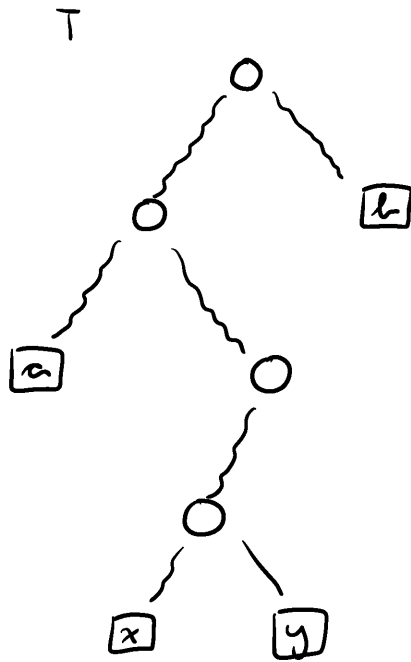
De ce nu așa?



Teorema 1 Fie \mathcal{C} un set de caractere și T arborele corespunzător codului optim. Fie $a, b \in \mathcal{C}$ caracterele de **frecvență minimă**.

Atunci a și b vor fi **frate** în arbore și vor fi pe **nivelul cel mai de jos**.

Dem Pr. prin R.A.



nodurile în
adâncime maximă

Construim un arbore T' în care inter schimbăm
nodurile a și x (unul din nodurile în adâncime max)

Presupunem $f(a) < f(x)$

$$\text{wst}(T) = \sum_{v \in V} f(v) \cdot d(v)$$

↙ depth

$$\text{Aratăm } \text{wst}(T') - \text{wst}(T) \leq 0$$

#

$$\text{wst}(T') - \text{wst}(T) = f(a) \cdot d_{T'}(a) + f(x) \cdot d_{T'}(x) -$$

$$f(a) \cdot d_T(a) - f(x) \cdot d_T(x)$$

$$= f(a) \left(\overset{d_{T'}(a)}{d_{T'}(a)} - d_T(a) \right) + f(x) \left(\underset{d_T(a)}{d_{T'}(x)} - d_T(x) \right)$$

$$= \left(d_T(x) - d_T(a) \right) \left(f(a) - f(x) \right)$$

≥ 0

≤ 0

Repetăm procedura cu x și y și obținem
arborele din teoremă

Teorema 2 Fie \mathcal{C} un set de caractere și
 T arborele corespunzător codului optim. Fie $x, y \in \mathcal{C}$
caracterele de **frecvență minimă**.

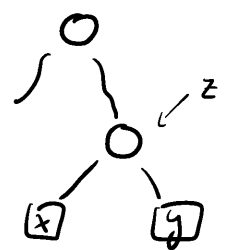
Fie $\mathcal{C}' = \mathcal{C} - \{x, y\} \cup \{z\}$ unde
 $f(z) = f(x) + f(y)$

Fie T' arborele optim pt \mathcal{C}' . Atunci T
obținut din T' în care înlocuim pe z cu un
nod intern ce are ca fiu pe x și y este
arbore optim pentru \mathcal{C}

$\mathcal{C}' \quad T'$



$\mathcal{C} \quad T$



Dem Pr. prin R.A. ca T nu e optim pt \mathcal{C} .
Dacă există T'' cu $\text{cost}(T'') < \text{cost}(T)$

Atunci există T''' cu
cu $\text{cost}(T''') < \text{cost}(T')$ un arbore cu cost
mai mic pt \mathcal{C}' (contradicție)

$$\begin{aligned} \text{wrt}(T) - \text{wrt}(T') &= \underbrace{f(a) \cdot d_{T'}(a)}_{d_{T'}(a)+1} + \underbrace{f(b) \cdot d_{T'}(b)}_{d_{T'}(b)+1} - \\ &\quad \underbrace{f(z) \cdot d_{T'}(z)}_{f(a)+f(b)} \end{aligned}$$

$$\begin{aligned} &= f(a) \cancel{d_{T'}(z)} - f(a) \cancel{d_{T'}(z)} + f(b) \cancel{d_{T'}(z)} \\ &\quad - f(b) \cancel{d_{T'}(z)} + f(a) + f(b) \end{aligned}$$

Dacă $\exists T''$ cu $\text{wrt}(T'') < \text{wrt}(T)$
 incluzind în T'' un alt decât nodul cu
 frecvență minimă cu sau fără o frecvență
 mică la

$$\Rightarrow T''' \text{ obținut astfel are } \text{wrt}(T''') = \text{wrt}(T'')^* - f(a) - f(b)$$

$$\text{wrt}(T''') < \text{wrt}(T) - f(a) - f(b) = \text{wrt}(T')$$