OPTIMALITATEA CODULUI HUFFMAN

Stan Bianca-Mihaela

Sa ne amintim mai intai ce era un cod Huffman...

- Codul Huffman este un mod de a comprima un set de date astfel incat sa avem 0 pierderi.

  Cum face asta codul Huffman?

- Folosind o codificare de tip “prefix-free”=niciun cod nu este prefix pentru un alt cod obtinut.
- exemplu:
  - {00, 01, 111} este o codificare in care niciun element nu este prefix pentru un alt element din sir
  - {00, 01, 011} nu respecta proprietatea de mai sus (01 este prefix pentru 011)

Cum se realizeaza aceasta proprietate utilizand codurile Huffman?

Sa luam un exemplu:

- Vrem sa codificam mesajul: BCCABBDDAECCBBAEDDCC.

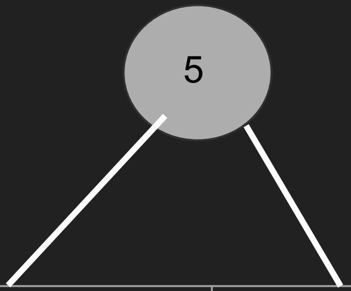
  Fiecare litera este retinuta prin codul sau ASCII, care este un cod de 8 biti.

  Avem 20 de litere => dimensiunea mesajului este 8*20=160 biti.

- Obesrvam ca vectorul de frecventa arata:

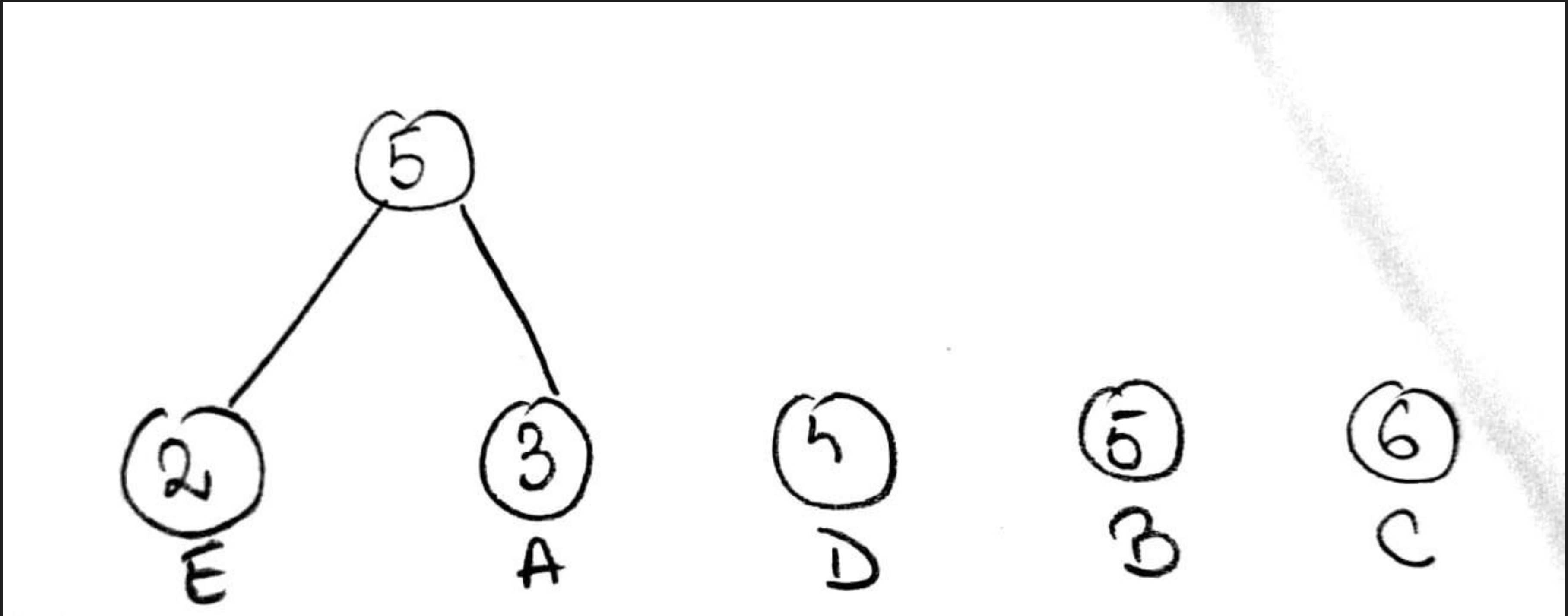
A	B	C	D	E
3	5	6	4	2

- Sortez literele in ordinea crescatoare dupa numaru de aparitii in mesaj.
- Urmaresc ceea ce se numeste “optimal merge parttern” care spune:
    - pun fiecare litera (reprezentata de numarul sau de aparitii) ca frunza intr-un arbore binar
    - la fiecare pas adaug in arbore un nou nod, format prin insumarea celor mai mici 2 noduri din arbore (nu luam in considerare nodurile care sunt deja fii)

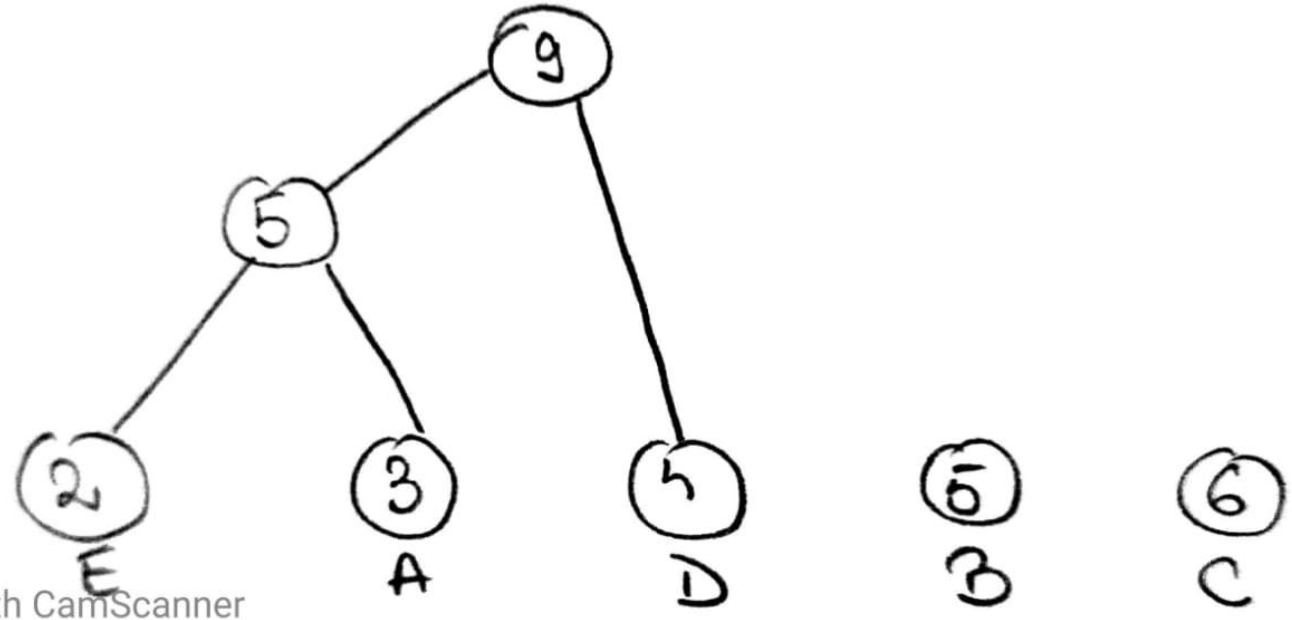


2	3	4	5	6
E	A	D	B	C

Pasul 1: cele mai mici frunze sunt 2 si 3 =>5

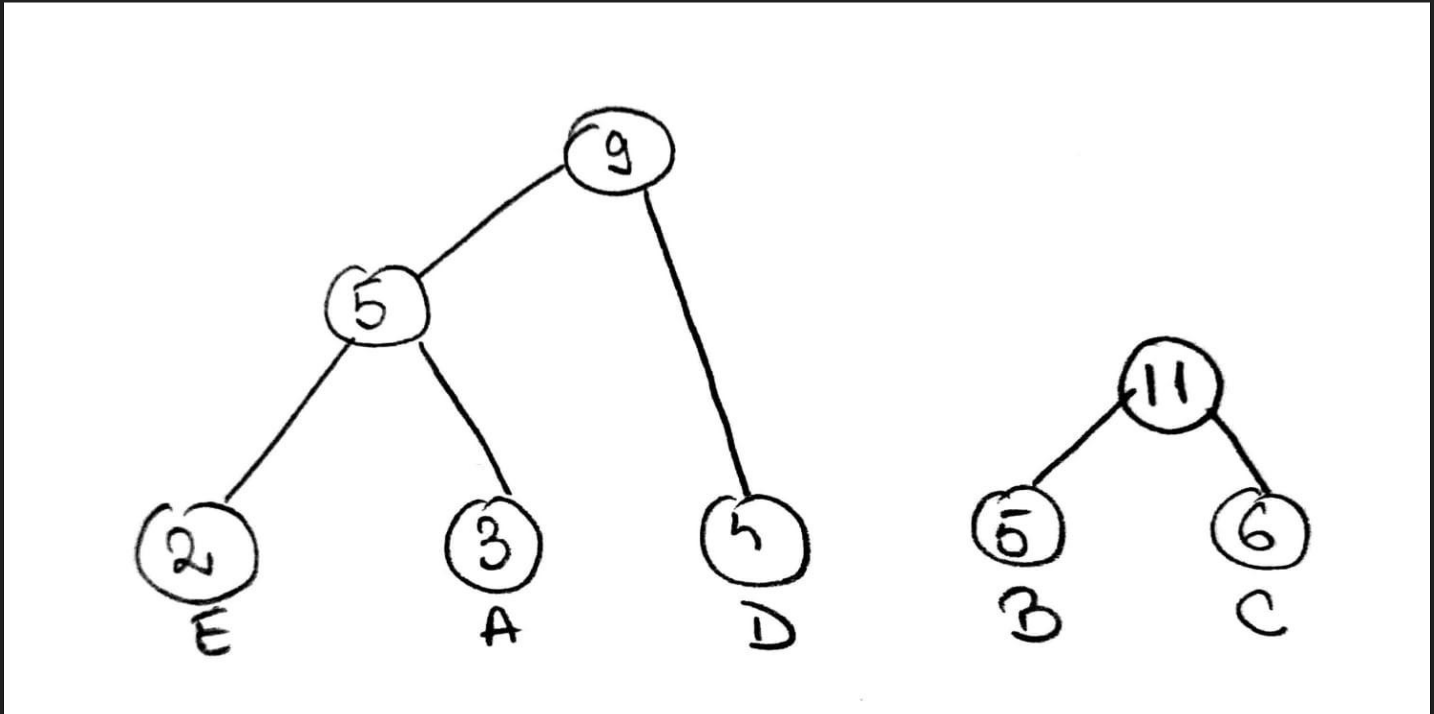


Pasul 2: cele mai mici frunze sunt 5 si 4 => 9

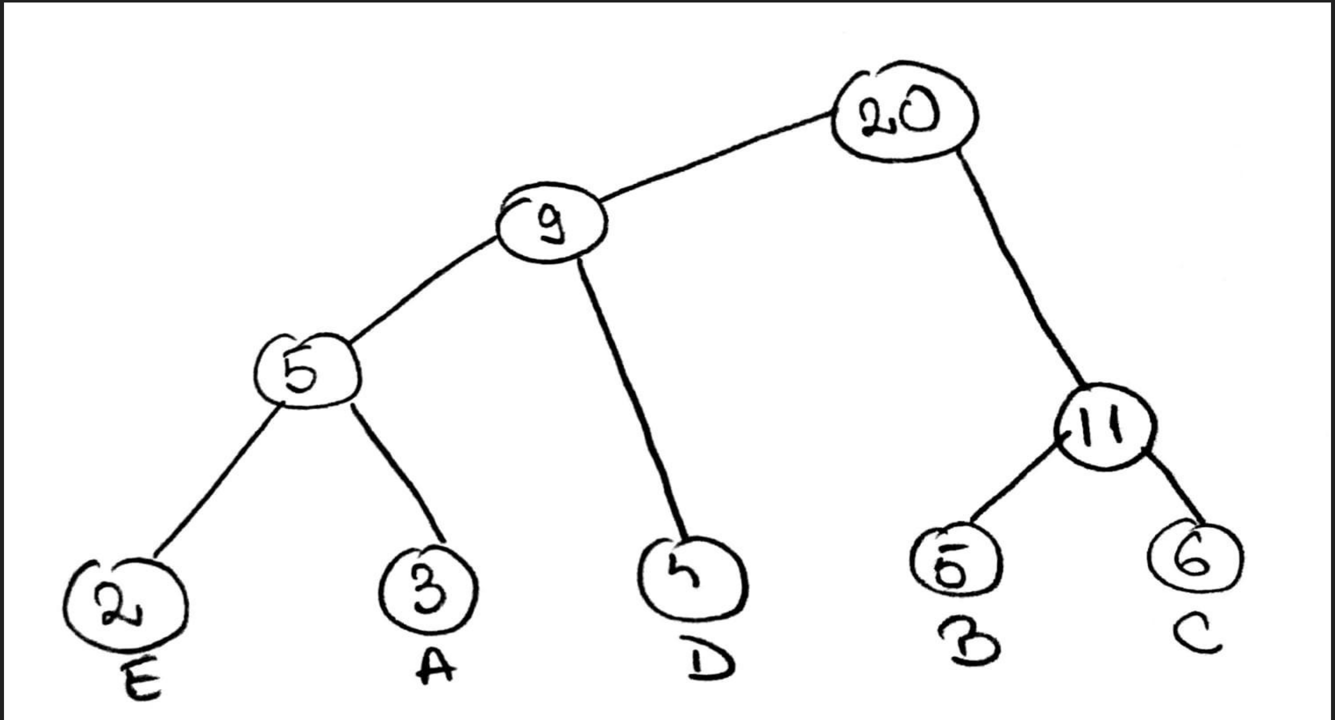


Scanned with CamScanner

Pasul 3: Cele mai mici noduri sunt acum 5 si 6 => 11

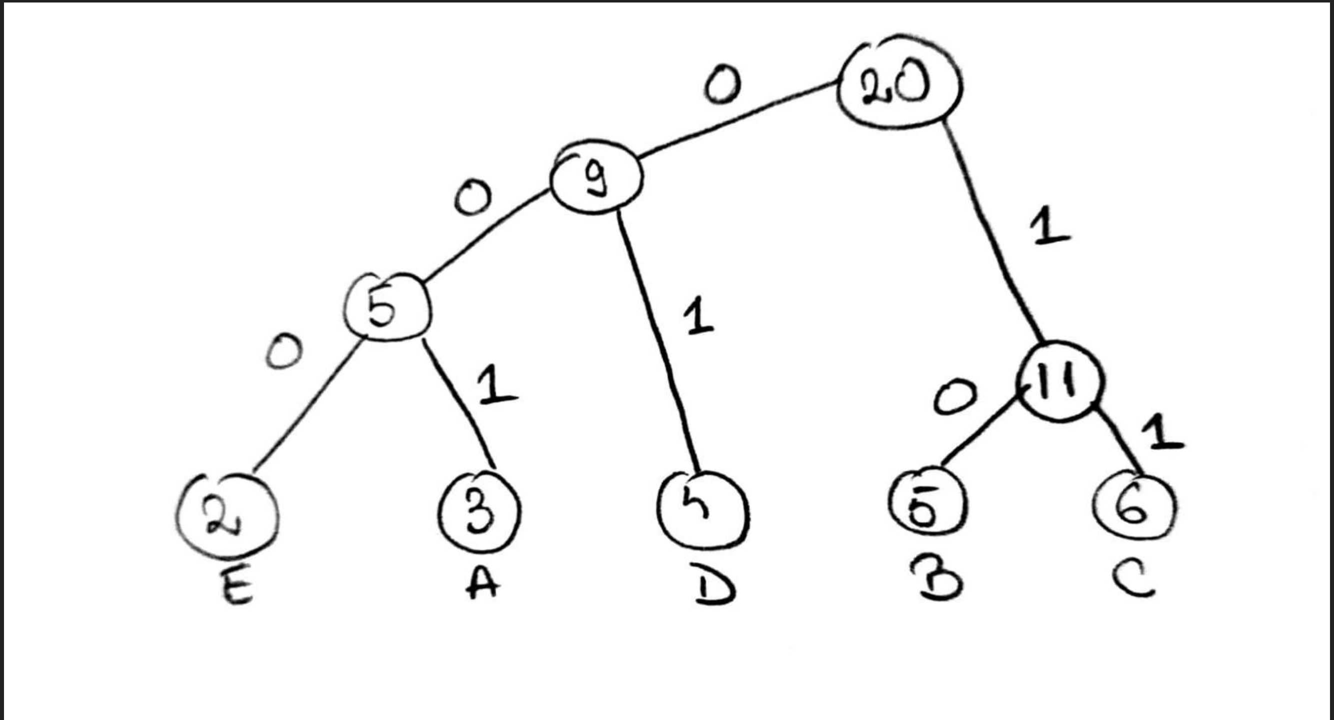


Pasul 4: cele mai mici noduri(care nu sunt parinti) sunt 9 si 11 =>20

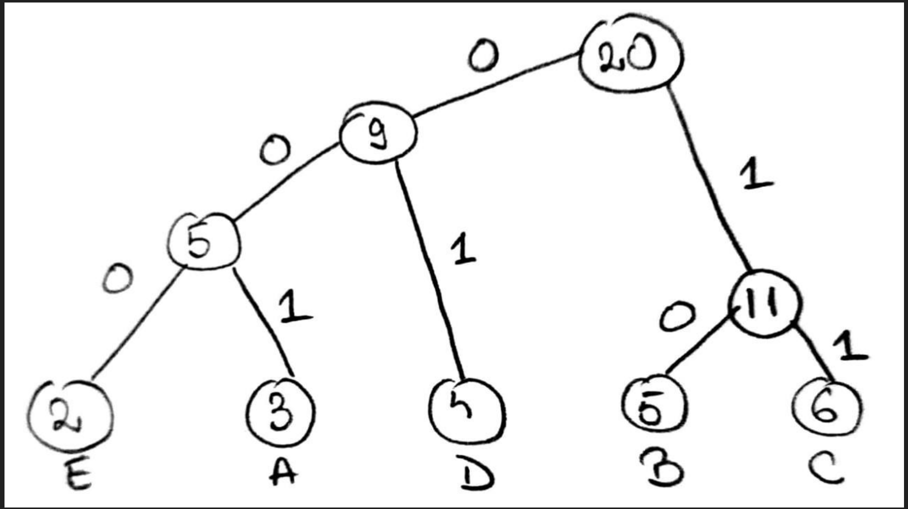


Pasul 5:

- legatura dintre tata si fiu drept o marchez cu 1
- legatura dintre tata si fiu stang o marchez cu 0



Pasul 6: codul pentru fiecare litera va fi format din literele intalnite in drumul din varful arborelui spre frunza corespunzatoare literei



	A	B	C	D	E
frecvență	3	5	6	4	2
cod	001	10	11	01	000

Care este dimensiunea actuala a mesajului?

	A	B	C	D	E
frecvență	3	5	6	4	2
cod	001	10	11	01	000
dimensiune	3b · 3 = 9	2b · 5 = 10	2b · 6 = 12	2b · 4 = 8	3b · 2 = 6

1b = 1 bit

=> dimensiunea totală a mesajului

= 9 + 10 + 12 + 8 + 6 = 45 biți

Imi trebuie totusi si un tabel pentru a putea decodifica mesajul. Ce dimensiune va avea tabelul?

A	B	C	D	E
001	10	11	01	000
3 biti	2 biti	2 biti	2 biti	3 biti

= 5 litere · 8 biti = 40 biti

$\underbrace{3 \text{ biti} + 2 \text{ biti} + 2 \text{ biti} + 2 \text{ biti} + 3 \text{ biti}}_{=\,12 \text{ biti}}$

40 + 12 = 52 de biti va avea tabelul

=> in final vom avea 45 + 52=97 biti

Se observa ca aceasta dimensiune este mult mai mica decat dimensiunea initiala a mesajului, care era de 160 de biti.

Este insa aceasta dimensiune minima? Cu alte cuvinte, este codul Huffman o modalitate de a obtine o codificare optima?

Vom demostra mai intai optimalitatea pe coduri binare, urmand sa generalizam foarte simplu din asta. Fie:

$B = 2$ ← alfabetul codului nostru este binar

$A = \{0, 1\}$ ← alfabetul

$X = \{x_1, x_2 \dots x_m\}$ ← alfabetul inițial, sursă

$|X| = m$, $m \in \mathbb{N}$

$P = (p_1, \dots p_m)$ pmf pe X

Ce e un pmf?

probability mass function: pentru x_k, p_k este probabilitatea ca alegând orice element din X, valoarea elementului să fie egală cu x_k.

ex: AABAC $\quad p_A = \dfrac{3}{5}$ ← nr de A-uri; ← nr total de litere.

$\ell = (\ell_1, \dots \ell_n)$ ← lungimile codurilor

Ce înseamnă că codul Huffman e optim?

Codifică mesajul corect și are o dimensiune minimă.

Am văzut în exemplul de mai sus că dimensiunea finală a mesajului este:

$$\underbrace{\sum_{k=1}^{n} \ell_k \cdot p_k}_{\text{mesajul}} + \underbrace{\sum \ell_k + n \cdot nr\_biti\_un\_caracter}_{\text{tabelul}}$$

Lema 1 (a ordonarii inverse)

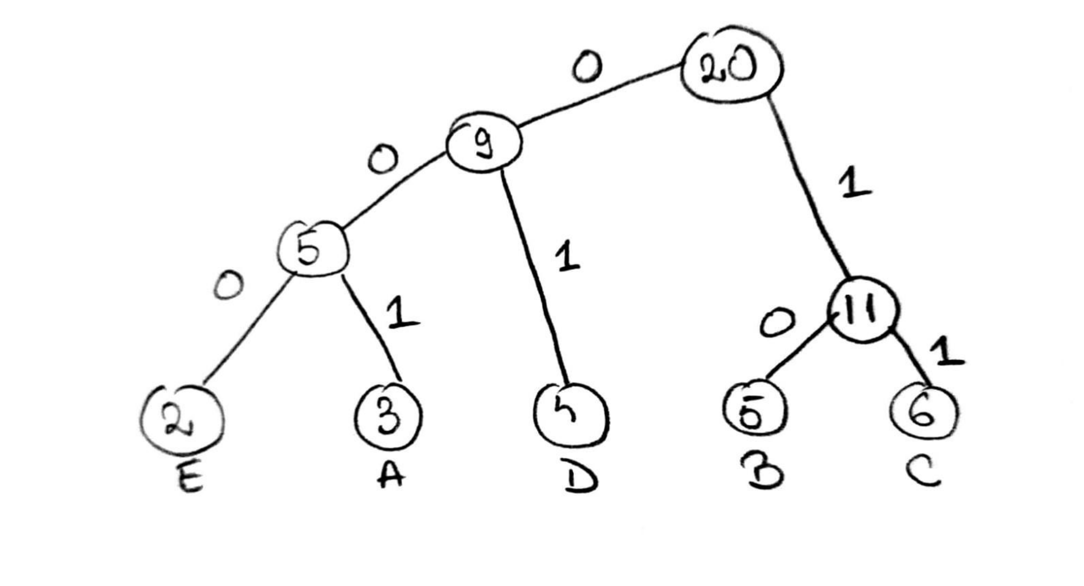
Intuitiv: Pentru orice cod optim de prefixe, lungimile codurilor sunt in ordinea inversa a probabilitatii fiecareui cod. Putem observa asta din tabel:

- cum C apare de 6 ori, codul sau are 2 cifre
- E apare doar de 2 ori, deci are 3 cifre

	A	B	C	D	E
frecvență	3	5	6	4	2
cod	001	10	11	01	000

Lema 1 (a ordonarii inverse)

Acest lucru devine intuitiv cand ne gandim la arbore. Vrem ca o litera utilizata des sa fie usor de accesat, adica aproape de varful arborelui.



Lema 1 (a ordonarii inverse)

Formal:

Pentru orice „prefix code" optim:

$\forall j, k$, if $p_j > p_k$, atunci $\ell_j \le \ell_k$

Demonstratie:

→ Fie C un „prefix code" optimal, cu lungimile ℓ_i ale codurilor $W = (w_1, \dots, w_m) \in A^*$.

ex: $w_2 = 01101$

→ Stim că $p_j > p_k$.

→ Fie C' a.î.
- $w_i' = w_i \quad \forall i$ a.î. $i \neq j$, $i \neq k$
- $w_j' = w_k$
- $w_k' = w_j$

(față de C, doar inversez w_k cu w_j)

Cum C e optimal $\Rightarrow \underbrace{L_C}_{\text{lungimea codului lui } C} \leq \underbrace{L_{C'}}_{\text{lungimea codului lui } C'}$

$\Rightarrow 0 \leq L_{C'} - L_C$

$\Leftrightarrow 0 \leq \sum_{i=1}^{m} e_i' p_i - \sum_{i=1}^{m} e_i p_i$

evident: $\left| \begin{array}{l} e_i' = e_i \quad \forall i \neq j, k \\ e_j' = e_k \\ e_k' = e_j \end{array} \right.$ (doar am inversat e_k cu e_j)

$$0 \leq \sum_{\substack{i \neq j,k \\ i=1}}^{m} e_i p_i + e_j' p_j + e_k' p_k - \sum_{i}^{m} e_i p_i$$

$$\Leftrightarrow 0 \leq \underbrace{e_j'}_{= e_k} p_j + \underbrace{e_k'}_{= e_j} p_k - e_j p_j - e_k p_k$$

$\Leftrightarrow \ 0 \leq e_k p_j + e_j p_k - e_j p_j - e_k p_k$

$\Leftrightarrow \ 0 \leq e_k (p_j - p_k) + e_j (p_k - p_j)$

$\Leftrightarrow \ 0 \leq (e_k - e_j)(p_j - p_k)$

știm că $p_j > p_k \Rightarrow p_j - p_k > 0$

$\Rightarrow \ 0 \leq e_k - e_j$

$\Leftrightarrow \ e_j \leq e_k$

Lema 2 (lema slaba a fratilor)

Formal:

Pentru orice pmf $p = (p_1, p_2 \dots p_m)$ există un „prefix code" optimal <u>BINAR</u> cu 2 frați.

Ce sunt 2 <u>frați</u> în acest context?

Două codări cu lungime maximă aî:

a) au aceeași lungime

b) se diferențiază doar prin ultimul bit

exemplu:

000
001
⋮
10100
11101
10110
10111

mi ordonat după lungime

aștia 2 sunt frați

Lema 2 (lema slaba a fratilor)

Demonstrație:

Fie C un „prefix code" optim a.î. $\sum l_i$ e minimal.

Demonstrăm a): Dacă dintre cele mai lungi coduri au aceeași lungime.

Presupunem că a) nu e adevărat.

$\Leftrightarrow$ Există un cod care e mai lung decât toate celelalte.

Să zicem că lungimile sunt ordonate:

$l_1 \le \dots < l_m$

cel mai lung e strict mai mare ca celelalte

Dacă notăm codurile din C astfel:

$$W = (w_1, \dots w_{m-1}, w_m)$$

putem să construim C' a.î.:

$$W' = (w_1, \dots w_{m-1}, w_m')$$

unde
- $\underbrace{w_m' \alpha}_{} = w_m, \alpha \in A$

  w_m' concatenat cu o literă din alfabetul A

- $l_m' = l_m - 1$

Lema 2 (lema slaba a fratilor)

Este oare acest c' nou construit, un „prefix-code"? Verifică proprietatea ca orice cod din c' să nu fie prefix pentru un alt cod din c'?

Evident, primele $n-1$ coduri respectă proprietatea pentru că sunt identice cu codurile din c.

Se pune problema pt w_n'.

Să spunem că w_n' e prefix pt alt cod din W', să îi spunem w_k.

$\Rightarrow$ are aceeași lungime w_k

(w_n era strict mai lung ca toate)

- w_n' prefix pt w_k
- w_n are aceeași lungime cu w_k

$\Rightarrow w_n' = w_k'$

dar asta înseamnă că w_k era prefix pt w_n (contradicție cu proprietatea lui c)

Lema 2 (lema slaba a fratilor)

$\Rightarrow$ C' este un „prefix code"

Cum optimalitatea este dată de $\sum e_i p_i$ și am micșorat suma totală a lungimilor $\Rightarrow$ $\sum e_i' p_i \leq \sum e_i p_i$

$$e_1 p_1 + \ldots + e_m' p_m \leq e_1 p_1 + \ldots + e_m p_m$$

e posibilă și egalitatea pentru că p_m poate fi 0, caz în care nu am modificat deloc optimalitatea

Dar spusesem în definiția lui c că $\sum e_i$ e minimală (motiv pentru care $\sum e_i p_i$ era minimă).

Dar $\sum e_i' = \sum e_i - 1$ $\Rightarrow$ contradicție asupra optimalității lui c

$\Rightarrow$ a) e adevărat

Lema 2 (lema slaba a fratilor)

Demonstrăm b):

Presupunem că b) e falsă.

$\Rightarrow$ Fie k lungimea maximă a codurilor.

$\Rightarrow$ orice 2 coduri de lungime k diferă undeva în primele $k-1$ cifre.

Ce vom face? Pentru:

000
001
⋮
1000
1010~~0~~
11101
10010
10111

Luăm unul dintre codurile de lungime maximă și îi tăiem ultimul bit.

$\Downarrow$

- noul cod va fi optimal
- vom verifica că are propr. de prefix

Formal: Avem C-ul nostru cu $W = (w_1, \dots w_{n-1}, w_n)$ ordonate a.î. $l_1 \le \dots \le l_n$

Construiesc C' cu $W' = (w_1, \dots w_{n-1}, w_n')$

unde:
- $w_n' \alpha = w_n$, $\alpha \in A$
- $l_n' = l_n - 1$

Cum am văzut la demonstrația de la a), C' este optim.

Este însă „prefix-code"?

Lema 2 (lema slaba a fratilor)

La fel ca la a), știm că $w_1, \dots, w_{m-1}$ respectă proprietatea de prefix.

Trebuie doar să verificăm doar că:

- $w_{m'}$ nu e prefix pentru niciun alt cod
- niciun alt cod nu e prefix pentru $w_{m'}$

- este $w_{m'}$ prefix pentru un alt cod?

  → dacă e prefix pt unul din codurile $\underbrace{\text{mai lungi decât el}}_{\text{lungime } k}$

  $\Rightarrow$ cele $k-1$ cifre ale lui $w_{m'}$ sunt identice cu primele $k-1$ cifre din acel alt cod

  $\Downarrow$

  contradicție cu proprietatea că oricare 2 coduri de lungime maximă diferă în primele $k-1$ cifre.

  → dacă e prefix pt. unul din codurile de lungime $k-1$, notat cu w_q

  $$\left|\begin{array}{l} \ell_{m'} = k-1 \\ \ell_q = k-1 \\ w_{m'} \text{ prefix al lui } w_q \end{array}\right. \Rightarrow w_{m'} \text{ identic cu } w_q \Rightarrow w_q \text{ e prefix pt } w_{m'}$$

  $\Downarrow$

  contradicție cu proprietatea de prefix a lui C

Lema 2 (a fratilor)

- niciun alt cod nu e prefix pt. w_{m}'?

  Fie $w_q \in W'$ a.î. w_q prefix pt w_m'.

  $\Rightarrow w_q$ prefix pt $w_m' \cdot \alpha$, $\alpha \in A$

  $\Leftrightarrow w_q$ prefix pt w_m

  $\Downarrow$

  contradictie cu proprietatea de prefix a lui C

$\Rightarrow$ C' este optim și are proprietatea de prefix

$\Rightarrow$ b) e adevărat

$\Rightarrow$ Lema 2 este adevărată

Lema 3 (lema puternica a fratilor)

Lema 3:

Fie p un pmf aî $p_1 \ge \dots \ge p_n$. Există un prefix code BINAR cu $W = (w_1, \dots w_n)$ aî:

a) $l_1 \le \dots \le l_{n-1} = l_n$

b) w_{n-1} și w_n diferă doar la ultimul bit

Lema 3 (lema puternica a fratilor)

Demonstrație:

Mă folosesc de lema slabă a fraților.

Fie c un prefix code optimal cu 2 frați, iar pmf-ul lui c este ordonat a.î. $p_1 \ge \dots \ge p_m$.

Demonstrez a):

Fie $p_k \ge p_{k+1}$.

$\Rightarrow$ fie: 1. $p_k > p_{k+1}$

2. $p_k = p_{k+1}$

Cazul 1: $p_k > p_{k+1}$

$$\xRightarrow{\text{Lema 1}} \ell_k \le \ell_{k+1}$$

Cazul 2: $p_k = p_{k+1}$

$\Rightarrow$ pot ordona p_k și p_{k+1} a.î.

$$\min(\ell_k, \ell_{k+1}) \le \max(\ell_k, \ell_{k+1})$$

$\Rightarrow$ pt $p_1 \ge \dots \ge p_m$, $\ell_1 \le \ell_2 \le \dots \le \ell_{m-1} \le \ell_m$

din lema slabă a fraților: $\ell_m = \ell_{m-1}$

$\Rightarrow \ell_1 \le \ell_2 \le \dots \le \ell_{m+1} = \ell_m$ c.c.t.d.

Lema 3 (lema puternica a fratilor)

Demonstrez b)

w_{m-1} și w_m sunt aceiași cu cei din lema slabă a fraților

=> w_{m-1} și w_m se diferențiază doar prin ultimul bit

=> Lema 3 este adevărată

Definitii

Fie $p = (p_1, \dots p_m)$, $p_1 \geq \dots \geq p_m$

construiesc $p' = (p_1, \dots, p_{m-2}, p_{m-1} + p_m)$

Definim:

- Huffman contraction: (notată cu HC)

Fiind dat un cod care respectă lema puternică a fraților, C^S, pentru pmf-ul p, atunci „Huffman contraction", HC, pentru p' a lui C^S are codurile:

$$\begin{cases} w_1^C = w_1^S \\ \vdots \\ w_{m-2}^C = w_{m-2}^S \\ w_{m-1}^C 0 = w_{m-1}^S \\ w_{m-1}^C 1 = w_m^S \end{cases}$$

exemplu: pentru

C^S
0
10
$w_{m-1}^S =$ 110
$w_m^S =$ 111

$\Rightarrow$

C^C
0
10
$11 = w_{m-1}^C$

(evident:

$w_{m-1}^C 0 = 110 = w_{m-1}^S$

$w_{m-1}^C 1 = 111 = w_m^S$)

Definitii

- Huffman extension (notată cu HE)

  Este procesul invers ca ce am făcut mai sus.

  => Fiind dat un cod optim C^0 pentru P', atunci HE (pentru P) este C^E cu codurile:

$$C^0 = (w_1^0, \dots, w_m^0) \Rightarrow C^E = (w_1^0, \dots, w_{m-1}^0, w_m^0 0, w_m^0 1)$$

exemplu:

C^0	C^E
0	0
10	10
11	110
	111

Proprietate

Proprietate:

Orice cod Huffman pentru p este HE a unui cod Huffman pentru p'.

(formalizează algoritmul de adăugare a nodurilor în arbore)

Lema 4 (lema extensiei)

Lema 4 (Lema extensiei)

Intuitiv: Dacă Huffman code-ul pentru p' e optim, atunci și codul pentru p e optim.

Formal: Fie p un pmf a.î. $p_1 \geq \dots \geq p_m$ și p' a.î. $p' = (p_1, \dots, p_{m-2}, p_{m-1} + p_m)$. HE (către p) a oricărui cod optim pentru p' este optim pentru p.

Demonstrație:

Fie C^O optim pentru p'.
Fie C^E HE a lui C^O către p.

exemplu:

C^O	C^E
0	0
10	10
11	110
	111

$$L^E = l_1^E p_1 + \dots + l_{m-1}^E p_{m-1} + l_m^E p_m$$

Știm că: $\bullet\ l_i^E = l_i^O,\ (\forall)\ i < m-1$; $\bullet\ l_{m-1}^E = l_m^E = l_{m-1}^O + 1$

$$\Rightarrow L^E = l_1^O p_1 + \dots + l_{m-2}^O p_{m-2} + (l_{m-1}^O + 1) p_{m-1} + (l_{m-1}^O + 1) p_m$$

$$\Leftrightarrow L^E = l_1^O p_1 + \dots + l_{m-2}^O p_{m-2} + l_{m-1}^O (p_{m-1} + p_m) + p_{m-1} + p_m$$

Știm că $L^O = l_1^O p_1 + \dots + l_{m-2}^O p_{m-2} + l_{m-1}^O (p_{m-1} + p_m)$

$$\Rightarrow L^E = L^O + p_{m-1} + p_m$$

Lema 4 (lema extensiei)

Fie C^S un cod care respectă lema puternică a fraților pe p (implică faptul că e optimal). Fie C^C HC (către p') a lui C^S.

$$L^C = e_1^C p_1 + \dots + e_{m-1}^C (p_{m-1} + p_m)$$

Știm că $e_i^C = e_i^S$, $(\forall)\, i < m-1$

$$e_{m-1}^C = e_{m-1}^S - 1 = e_m^S - 1$$

$$\Rightarrow L^C = e_1^S p_1 + \dots + e_{m-2}^S p_{m-2} + (e_{m-1}^S - 1) p_{m-1} + (e_m^S - 1) p_m$$

Știu că $L^S = e_1^S p_1 + \dots e_{m-1}^S p_{m-1} + e_m^S p_m$

$$\Rightarrow L^C = L^C - p_{m-1} - p_m$$

Optimalitatea codurilor Huffman

În demonstrație ne vom referi la coduri Huffman standard, adică obținute prin combinarea celor mai mici elemente din pmf-ul dat.

Ce e un pmf?

$$P_x = \frac{\text{nr apariții ale lui } x \text{ în mesaj}}{\text{nr de litere din mesaj}}$$

Vrem să demonstrăm că orice cod Huffman este optim.

$\Updownarrow$

① Orice cod Huffman pe un pmf cu K elemente este optim.

Am văzut din primul exemplu că optimalitatea este dată de minimizarea $\sum l_i p_i$.

Vom face o demonstrație prin inducție.

Pasul 1 (verificarea)

pentru $K = 2 \Rightarrow P' = (p_1 + p_2)$ optim

Optimalitatea codurilor Huffman

Pasul 2 (inducție)

Presupun că ① este valabilă pt $k = n-1$.

Demonstrez că ① este valabilă pt $k = n$.

→ Fie g un pmf arbitrar cu n elemente.

→ Fie p pmf-ul g sortat descrescător, $p = (p_1, p_2 \dots p_n)$

→ Fie c un cod Huffman pe p.

Dacă $p' = (p_1, \dots, p_{n-1} + p_n)$ și C' este un cod Huffman pt p'.

$\xRightarrow{\text{ipoteză}}$ C' este optim

Se vede că C este HE lui C'.

Folosim lema extensiei:

INTUITIV: Dacă Huffman codul pt p' este optim, atunci și Huffman cod-ul pt p este optim.

⇒ C este optim ⇒ am demonstrat ① pt n elemente

$$\vdots$$

$$C_n(x_n) = 111\cdots 10.$$

For $i = 1, \dots, k$: Let $m_i = n/p_i$. So if $l_i \geq m_i$ then $l_i p_i \geq n$.

Let $\mathcal{C} = \{ C : C$ is U.D. with lengths $l_1, \dots, l_n$ where $l_i \leq m_i$ for $i = 1, \dots, k \}$.

$\Rightarrow C_n \in \mathcal{C}$ since $\hat{l}_i \leq n = m_i p_i \leq m_i \quad \forall i = 1, \dots, k.$

Lemma (Existence): For any pmf p, there exists an optimal prefix code.

Lemma:

<u>Lemma</u> (Siblings): For any pmf $p=(p_1,\dots,p_n)$, there exists an optimal binary prefix code s.t. two of the longest codewords:

(a) have the same length

and (b) differ only the last bit.

$\# L = \sum l_i p_i$

e.g.

000

001

⋮

1000

10100

10101

10110

101110

<u>Proof</u>: Let C be opt. prefix s.t. $\sum l_i$ is minimal.

(a) Suppose (a) is not true $\Rightarrow l_1 \le \dots < l_n$

$\underline{C}$

w_1

⋮

w_{n-1}

w_n

Key Lemma

Lemma (Sibling code): Let p be a pmf s.t. $p_1 \geq \cdots \geq p_n$.

There exists an optimal binary prefix code with codeword $w_1, \ldots, w_n$ $(w_i = C(x_i))$ s.t.

ⓐ $l_1 \leq \cdots \leq l_{n-1} = l_n$

and ⓑ w_{n-1} and w_n differ only in the last bit.

Defn: Given a sibling code C^s for p, the H-contraction C^c to p' is

1.0 1 .5 1

0
10
110
111

$p_1 \geq \cdots \geq p_{n-1} \geq p_n$

0 1

etc $\quad p_{n-1} + p_n$

C^c	C^s
0	0
10	10
	110
	111

Key Lemma

Lemma (Sibling code): Let p be a pmf s.t. $p_1 \geq \cdots \geq p_n$.
There exists an optimal binary prefix code with codeword $w_1, \ldots, w_n$ $(w_i = C(x_i))$ s.t.

$\left.\begin{array}{l} \text{(a) } \ell_1 \leq \cdots \leq \ell_{n-1} = \ell_n \\ \text{and (b) } w_{n-1} \text{ and } w_n \text{ differ only in the last bit.} \end{array}\right\}$ Sibling code

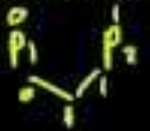
	0	10	110	111
Let $p = (p_1, \ldots, p_n)$, $p_1 \geq \cdots \geq p_n$.	.5	.3	.1	.1

Lemma (Extension): Let p be a pmf s.t. $p_1 \geq \cdots \geq p_n$. Let $p' = (p_1, \ldots, p_{n-2}, p_{n-1} + p_n)$. The H-extension (to p) of any optimal code for p' is optimal for p.

Theorem: Any Huffman code is optimal.

Pf: $(*_k)$ Any Huffman code on a pmf with k elements is optimal.

(Base) $(*_2)$ is true.



(Ind.) Suppose $(*_{n-1})$ is true. Let q be a pmf on n elements