(1) Siano X, Y due v.a. sullo stesso alfabeto $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5\}$ con due d.p. p_X, p_Y rispettivamente.

Consideriamo due funzioni di codifica $\varphi_1(\cdot), \varphi_2(\cdot)$ per gli elementi di $\mathcal{A}$:

x	$p_X(x)$	$p_Y(x)$	$\varphi_1(x)$		$\varphi_2(x)$		q	
a_1	$1/2$	$1/2$	0	(1)	0	(1)	$2^{-1}/\alpha$	$1/2$
a_2	$1/4$	$1/8$	10	(2)	100	(3)	$2^{-3}/\alpha$	$1/8$
a_3	$1/8$	$1/8$	110	(3)	101	(3)	$2^{-3}/\alpha$	$1/8$
a_4	$1/16$	$1/8$	1110	(4)	110	(3)	$2^{-3}/\alpha$	$1/8$
a_5	$1/16$	$1/8$	1111	(4)	111	(3)	$2^{-3}/\alpha$	$1/8$

$\alpha = 1/2 + 1/8 + 1/8 + 1/8 + 1/8 = 1$

(1) Calcolare $H(X), H(Y), D(p_X \| p_Y)$ e $D(p_Y \| p_X)$

(2) Mostrare che la lunghezza media delle parole di codice di φ_1 utilizzata per X coincide con $H(X)$ e quindi φ_1 è ottimale per X. Mostrare che φ_2 è ottimale per Y.

(3) Si supponga di utilizzare la codifica $\varphi_2(\cdot)$ per X.

Che lunghezza media otteniamo per le parole di codice? Di quanto è maggiore di $H(X)$? Si evidenzi la relazione tra la risposta e il valore di $D(p_X \| p_Y)$.

$p_X = (1/2, 1/4, 1/8, 1/16, 1/16)$
$p_Y = (1/2, 1/8, 1/8, 1/8, 1/8)$

(1) $H(X) = 1/2 \cdot 1 + 1/4 \cdot 2 + 1/8 \cdot 3 + 2/16 \cdot 4 = 15/8 = \boxed{2 - 1/8}$ bit.

$H(Y) = 1/2 \cdot 1 + 4/8 \cdot 3 = 1/2 + 3/2 = 2$ bit.

$\longrightarrow D(p_X \| p_Y) = \sum_i p_X(x_i) \log \frac{p_X(x_i)}{p_Y(x_i)} = \cancel{1/2 \log 1} + 1/4 \log 2 + \cancel{1/8 \log 1} + 2/16 \log 1/2$
$= 1/4 + 1/8 (-1) = 1/8$.

$D(p_Y \| p_X) = 1/8$.

(2) $\longrightarrow E[L^{(1)}] = 1/2 \cdot 1 + 1/4 \cdot 2 + 1/8 \cdot 3 + 1/16 \cdot 4 + 1/16 \cdot 4 = 15/8 = 2 - 1/8 = H(X)$. φ_1 ottimale per X

$E[L^{(2)}] = 1/2 \cdot 1 + (1/2) \cdot 3 = 1/2 + 3/2 = 2$ bit $= H(Y)$. φ_2 ottimale per Y.

$(1/8 + 1/8 + 1/8 + 1/8) \cdot 3$

$L^{(3)}$: lunghezza di una parola di codice ottenuta applicando $\varphi_2(\cdot)$ a simboli generati secondo la d.p. p_X

$$E[L^{(3)}] = \frac{1}{2}\cdot 1 + \underbrace{\frac{1}{2}}_{(1/4+1/8+1/16+1/16)}\cdot 3 = 2 \;\; > \;\; H(X) = \frac{15}{8} = 2 - \frac{1}{8}$$

Poiché $E[L^{(3)}] > E[L^{(2)}]$, $\varphi_2(\cdot)$ non è ottimale per X

Ridondanza : $E[L^{(3)}] - H(X) > 0$ $\qquad E[L^{(3)}] - H(X) = 2 - (2 - 1/8) = 1/8$.

$\forall$ codifica : $E[L] \geq H(X)$ $\leftarrow p$ $\qquad$ Nel caso binario $(D=2)$ $\qquad \varphi_2$

$\{l_i\} \rightsquigarrow q_i = \frac{2^{-l_i}}{\alpha}$ $\qquad \boxed{\left(\alpha = \sum_{i=1}^{k} 2^{-l_i}\right)}$

$$D(p\|q) = \sum_i p_i \log \frac{p_i}{2^{-l_i}}\cdot\alpha = \overbrace{E[L] - H(p)}^{\text{Ridondanza}} + \log(\alpha)$$

Caso binario : $\quad$ Ridondanza $= D(p\|q) - \log\alpha \overset{(\alpha\le 1)}{\geq} D(p\|q)$ $\qquad \boxed{\text{Se } \alpha = 1, \text{ Ridondanza} = D(p\|q)}$

(2) Shannon - Fano → non è ottimale
Fano → non è ottimale
Huffman → è ottimale

Es. $\mathcal{A} = \{a, b, c, d, e, f\}$

$$p = (0.13, 0.1, 0.03, 0.15, 0.18, 0.41)$$

28	a →	13	/100
		10	/100
	c →	3	/100
21	d →	15	/100
	e →	18	/100
		41	/100

$\{41, 3\}$

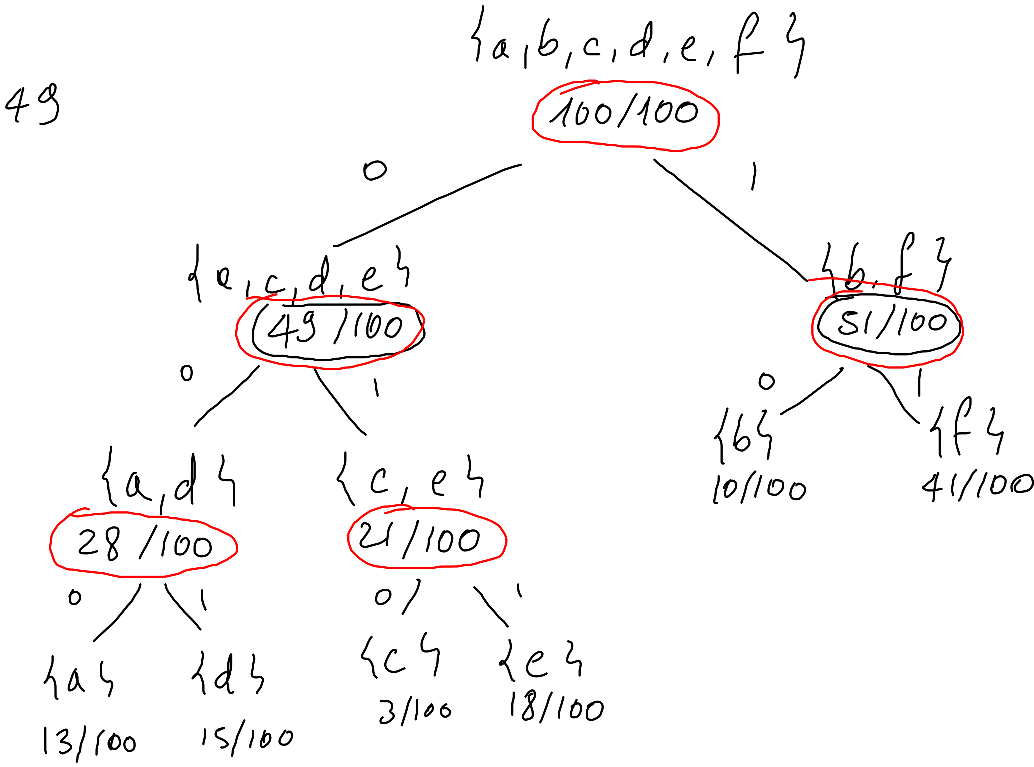
Fano: $Pr(\{a, c, d, e\}) = 0.13 + 0.03 + 0.15 + 0.18 = 0.49$

$Pr(\{b, f\}) = 0.1 + 0.41 = 0.51$

	$\varphi(\cdot)$	
a	000	3
b	10	2
c	010	3
d	001	3
e	011	3
f	11	2

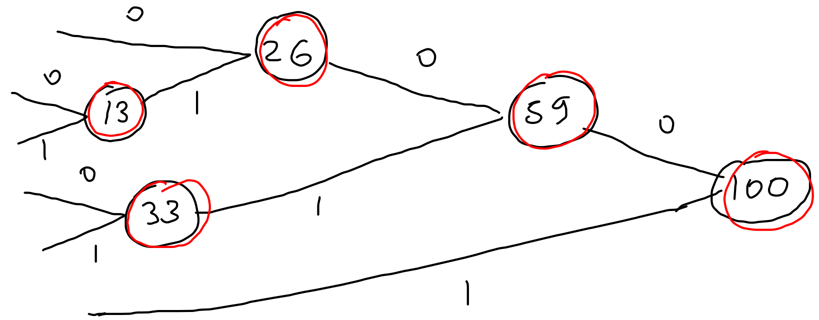
$$E[L^{(Fano)}] = 3 \times \left(\frac{49}{100}\right) + 2 \times \left(\frac{51}{100}\right) = 2.49$$

$$\frac{28 + 21 + 49 + 51 + 100}{100} = \frac{249}{100} = 2.49$$



Huffman

$\ell(x)$	$\varphi(x)$	x		
3	000	a	13	(/100)
4	0010	b	10	(/100)
4	0011	c	3	"
3	010	d	15	"
3	011	e	18	"
1	1	f	41	"



$$E[L^{(\text{Huffman})}] = \frac{13}{100}\cdot 3 + \frac{10}{100}\cdot 4 + \frac{3}{100}\cdot 4 + \frac{15}{100}\cdot 3 + \frac{18}{100}\cdot 3 + \frac{41}{100}\cdot 1 = 2.31 < 2.49$$

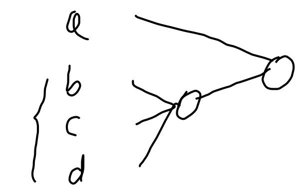
$$= \frac{13+33+26+59+100}{100} = \frac{231}{100} = 2.31.$$

3. Codifica di Huffman nel caso D-ario ($D > 2$)

$D = 3 \quad K = 4$

Aggreghiamo ad ogni passo i D simboli meno probabili.

a
b
c
d



$-D+1$
$-(D-1)$

Dà un codice ottimo solo se l'albero di codifica è completo:

ogni nodo ha o 0 o D figli; questo accade se e solo se $\boxed{K \overset{(*)}{=} D + j(D-1)}$, per qualche j intero non negativo

$4 = 3 + j(3-1)$
$3 + j\,2$

Se (*) non è soddisfatta, aggiungiamo simboli fittizi a probabilità zero in modo da soddisfarla.

$7 = 3 + 2 \cdot 2$

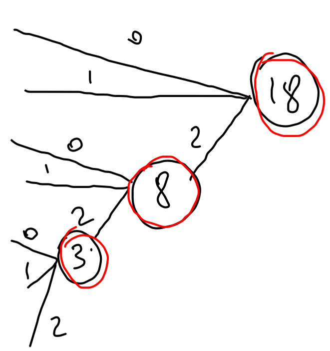
Es. $A = \{a, b, c, d, e, f\}$ ($K = 6$), voglio $D = 3$

$6 = 3 + j(2)$ non è soddisfatta da nessun j intero

$p = \left(\frac{6}{18}, \frac{4}{18}, \frac{3}{18}, \frac{2}{18}, \frac{2}{18}, \frac{1}{18}\right)$

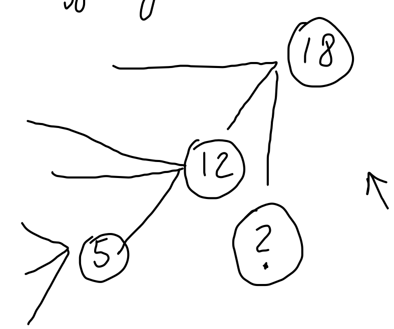
$\Rightarrow$ aggiungo 1 simbolo

$\varphi(\cdot)$		
0	a	6
1	b	4
20	c	3
21	d	2
220	e	2
221	f	1
(222)	x	0



$E[L] = \frac{18+8+3}{18} = \frac{29}{18}$

a	6
b	4
c	3
d	2
e	2
f	1



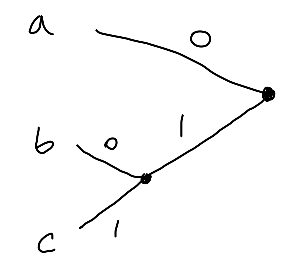
$\frac{18+12+5}{18} = \frac{35}{18}$

④ Quali di questi insiemi di parole di codice sono ottenibili dalla codifica di Huffman?

(a) $\{0, 10, 11\}$: sì

(b) $\{00, 01, 10, 110\}$

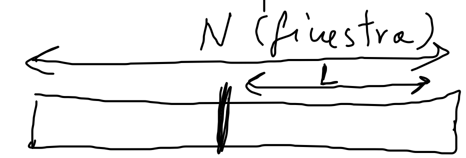
(c) $\{01, 10\}$



(b): NO perché non è ottimale: 110 può essere abbreviato in 11

(c): NO: si poteva utilizzare $\{0, 1\}$

⑤ Decomprimere la seguente sequenza in base 10 compressa col metodo Ziv-Lempel (LZ77)
(K=10)
con una finestra di lunghezza 14, con 7 simboli di lookahead.
(N=14) (L=7)



S $\xrightarrow{\text{compressione}}$ $\underbrace{\boxed{605}\ \boxed{651}\ \boxed{600}\ \boxed{640}}_{\text{blocchi}}$

Lunghezza blocchi: parte puntatore: $\lceil \log_{10}(N-L) \rceil = \lceil \log_{10} 7 \rceil = 1$

parte lunghezza: $\lceil \log_{10} L \rceil = \lceil \log_{10} 7 \rceil = 1$

parte nuovo simbolo: 1 $\quad$ 1

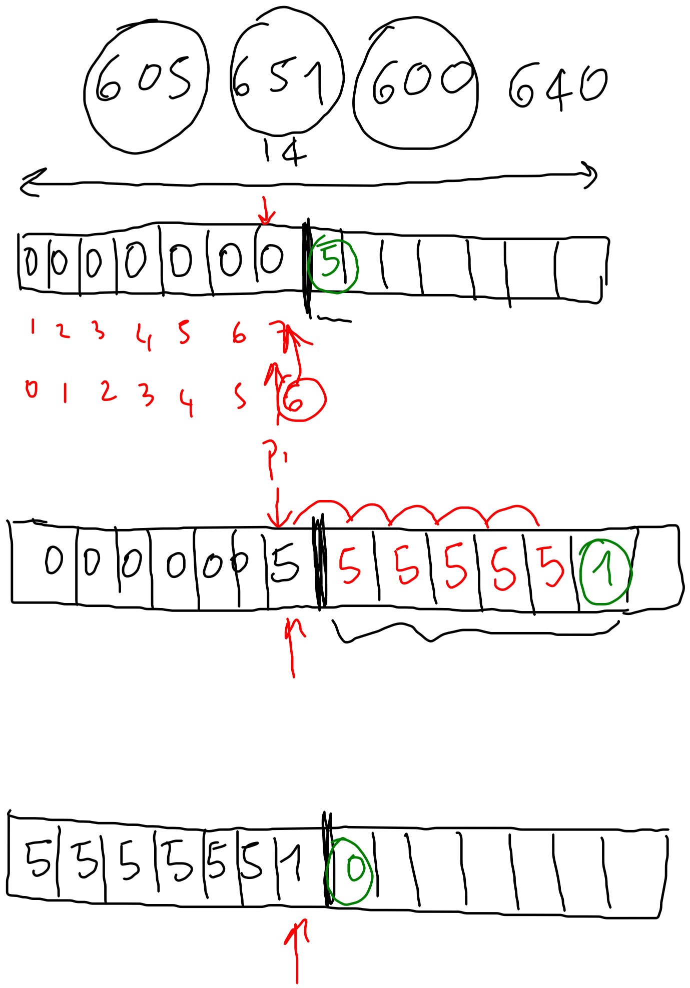
$\Big\}\ 3$

F_1 :

F_2 :

F_3 :

F_4 :



$N = 14$, $L = 7$

605

puntatore

0

nuovo simbolo

$p_1 = 7$

1° messaggio : 5

651

nuovo simbolo

$p_2 = 7$

2° messaggio : 555551

600

$p_3 = 7$

3° messaggio : 0

640

4° messaggio : 00000