CODICI DI HUFFMAN

- CONSENTONO FATTORI DI COMPRESSIONE TRA IL 20% E IL 90%

- **PROBLEMA**: TROVARE UNA CODIFICA DI UN FILE DI CARATTERI IN MODO DA MINIMIZZARNE LA DIMENSIONE

ESEMPIO: FILE DI 100 CARATTERI

CAR.	FREQ.	COD 1 (8 BIT)	COD 2 (3 bit)	COD 3	
a	45	00000000	000	0	45
b	13	00000001	001	101	39
c	12	00000010	010	100	36
d	16	00000011	011	111	48
e	9	00000100	100	1101	36
f	5	00000101	101	1100	20
	100	800 bit		300 bit	224 bit

Risparmio 62,5%

Risparmio 25,3%

72%

LUNGHEZZA FISSA

LUNGH. VARIABILE

~25% IN MENO

ES. a b a c

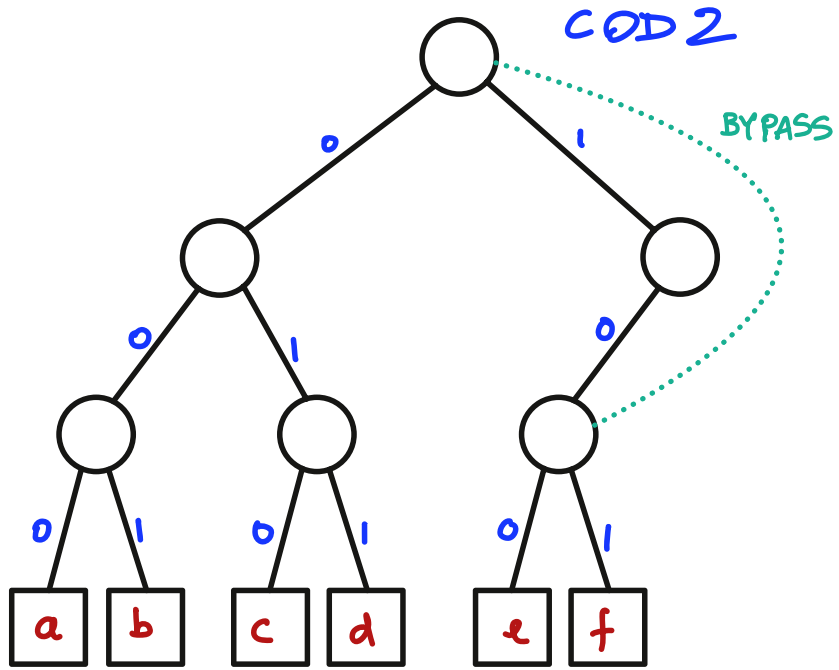
COD 2: 000 001 000 010

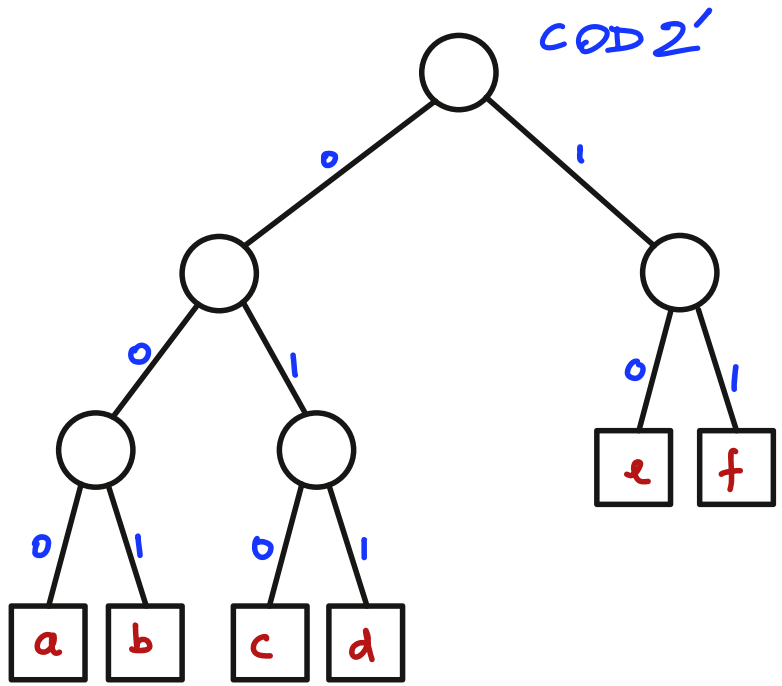
COD 3: 0 101 0 100

ALBERI DI DECODIFICA

CAR.	FREQ.	COD2	COD2'	
a	45	000	000	135
b	13	001	001	39
c	12	010	010	36
d	16	011	011	48
e	9	100	10	18
f	5	101	11	10
		300	286	286

RISPARMIO: $\frac{14}{300} \cdot 100 \simeq 4.67\,\%$

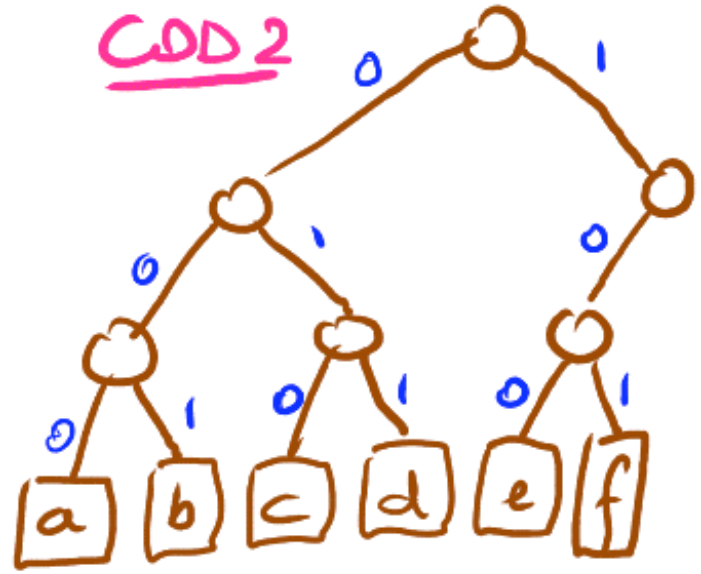




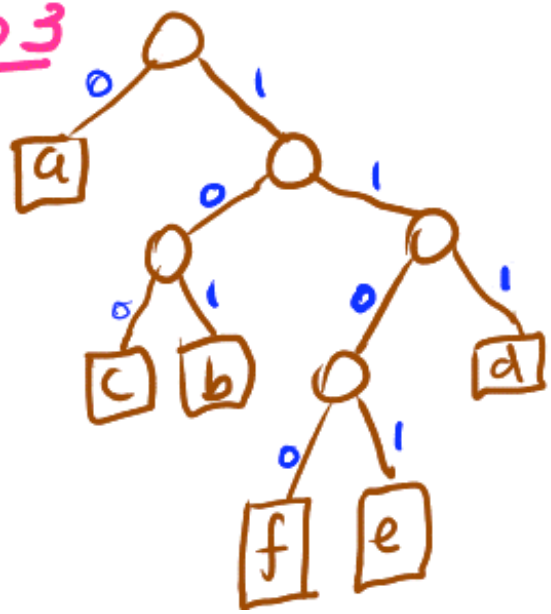
ALBERI DI DECODIFICA

CAR.	FREQ.	COD2 (3 bit)	COD3
a	45	000	0
b	13	001	101
c	12	010	100
d	16	011	111
e	9	100	1101
f	5	101	1100

COD 2



COD 3



ES.

a b a c

COD 2

000 001 000 010 — a b a c

COD 3

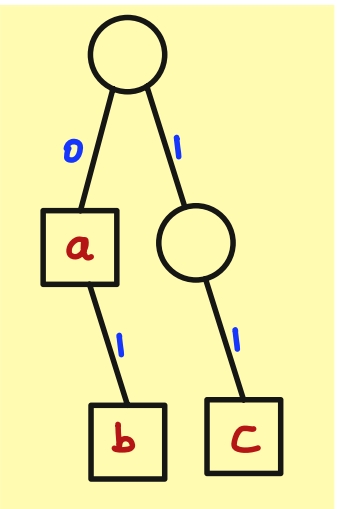
0 101 0 100 — a b a c

- CODICI PREFISSI: SONO CODICI IN CUI NESSUNA CODIFICA E' PREFISSO DI UN'ALTRA CODIFICA

ESEMPIO DI CODICE NON PREFISSO

a	0
b	01
c	11

0 11 11 11 11 11 11 11
b c c c a c



ESEMPIO DI CODICE NON PREFISSO AMBIGUO

a	0
b	1
c	01

01

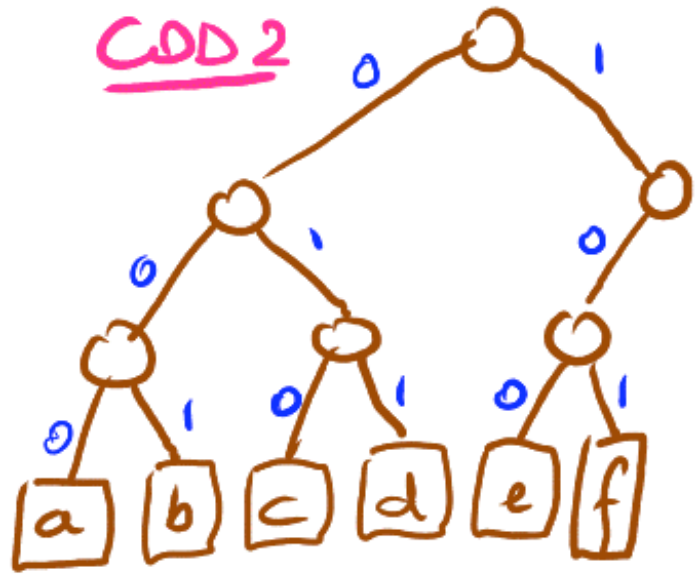
a b c

?

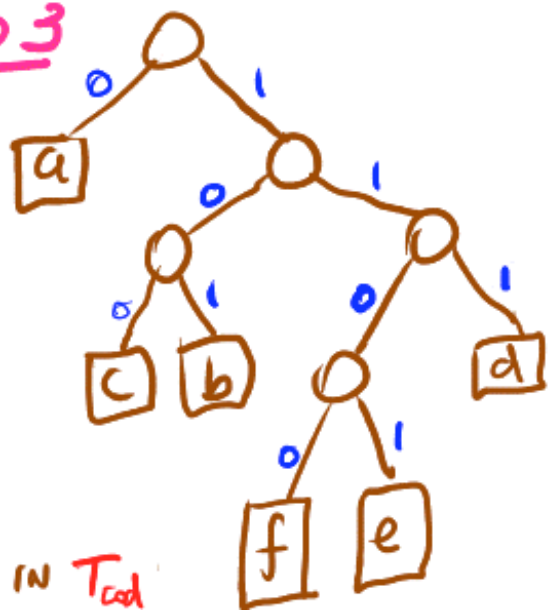
ALBERI DI DECODIFICA

CAR.	FREQ.	COD2 (3 bit)	COD3
a	45	000	0
b	13	001	101
c	12	010	100
d	16	011	111
e	9	100	1101
f	5	101	1100

COD 2



COD 3



COSTO DELLA CODIFICA:

$$B(cod) = \sum_{c \in C} f(c)\,|cod(c)|$$

$$= \sum_{c \in C} f(c)\, d_{T_{cod}}(c) =_{def} B(T_{cod})$$

T_{cod} : ALBERO DI DECODIFICA $\quad f: C \to \mathbb{N}$

C : ALFABETO $\quad d_{T_{cod}}$: PROFONDITÀ IN T_{cod}

$B(cod_3) = 45 \cdot 1 + 13 \cdot 3 + 12 \cdot 3 + 16 \cdot 3 + 9 \cdot 4 + 5 \cdot 4 = 45 + 41 \cdot 3 + 14 \cdot 4 = 45 + 123 + 56 = 224$

COSTO DEGLI ALBERI DI DECODIFICA

C : ALFABETO

$f : C \to \mathbb{N}$: FUNZIONE FREQUENZA

$cod : C \to \{0,1\}^+$: CODIFICA DEI CARATTERI DI C

T_{cod} : ALBERO DI DECODIFICA

$B(cod)$: COSTO DELLA CODIFICA DI UN TESTO NELL'ALFABETO C CON FREQUENZA f

$|cod(c)| \quad (c \in C)$: LUNGHEZZA DI $cod(c)$

$d_{T_{cod}}(c) \quad (c \in C)$: PROFONDITA' DELLA FOGLIA DI T_{cod} CONTENENTE IL CARATTERE c

SI HA:

$$B(cod) := \sum_{c \in C} f(c) \cdot |cod(c)| = \sum_{c \in C} f(c) \cdot d_{T_{cod}}(c)$$

PONIAMO

$$B(T_{cod}) := \sum_{c \in C} f(c) \cdot d_{T_{cod}}(c)$$

(COSTO DI T_{cod})

DUNQUE

$$B(T_{cod}) = B(cod)$$

PROBLEMA: TRA TUTTI GLI ALBERI DI DECODIFICA RELATIVI AD UN SISTEMA (C, f) (DOVE $f: C \to \mathbb{N}$) DETERMINARE QUELLO DI COSTO MINIMO, CIOÈ L'ALBERO BINARIO DI DECODIFICA T TALE CHE

$$B(T) = \sum_{c \in C} f(c) d_T(c)$$

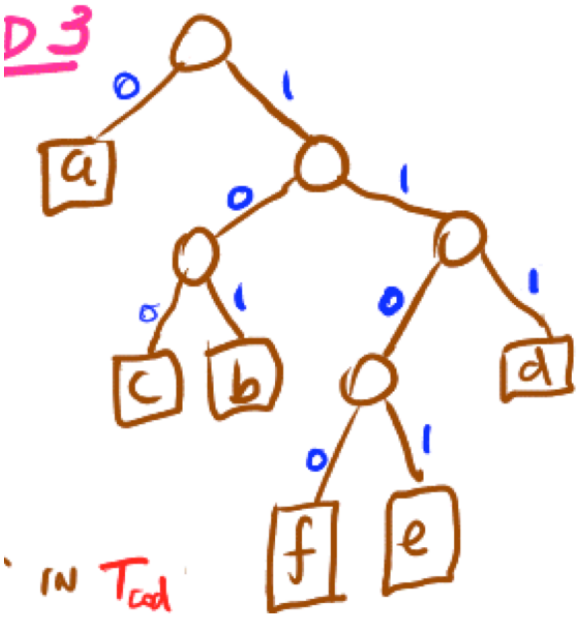
SIA MINIMO

NOTA: CIÒ CORRISPONDE A CERCARE UNA CODIFICA DI COSTO MINIMO RELATIVA AL SISTEMA (C, f)

OSSERVAZIONE: POSSIAMO LIMITARE LA NOSTRA RICERCA AGLI ALBERI BINARI PIENI, QUELLI CIOÈ PRIVI DI NODI INTERNI CON UN SOLO FIGLIO.

OSSERVAZIONE: IL NUMERO DI NODI INTERNI IN UN ALBERO BINARIO PIENO CON m FOGLIE È m-1.

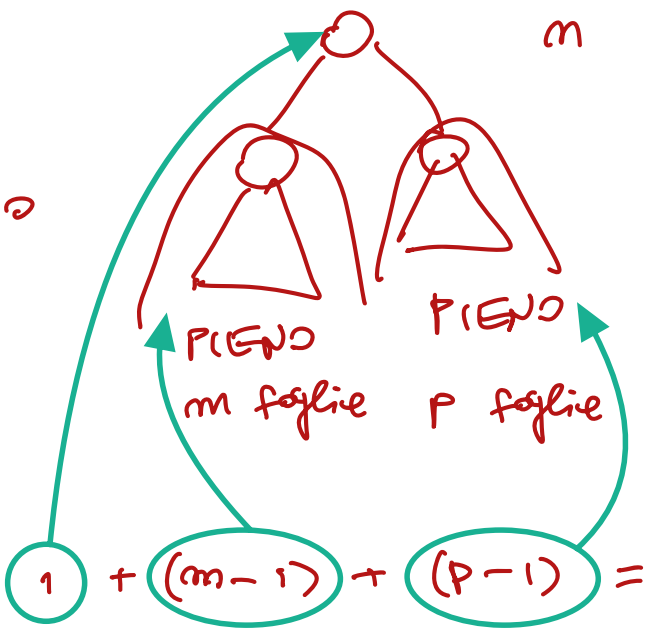
D3



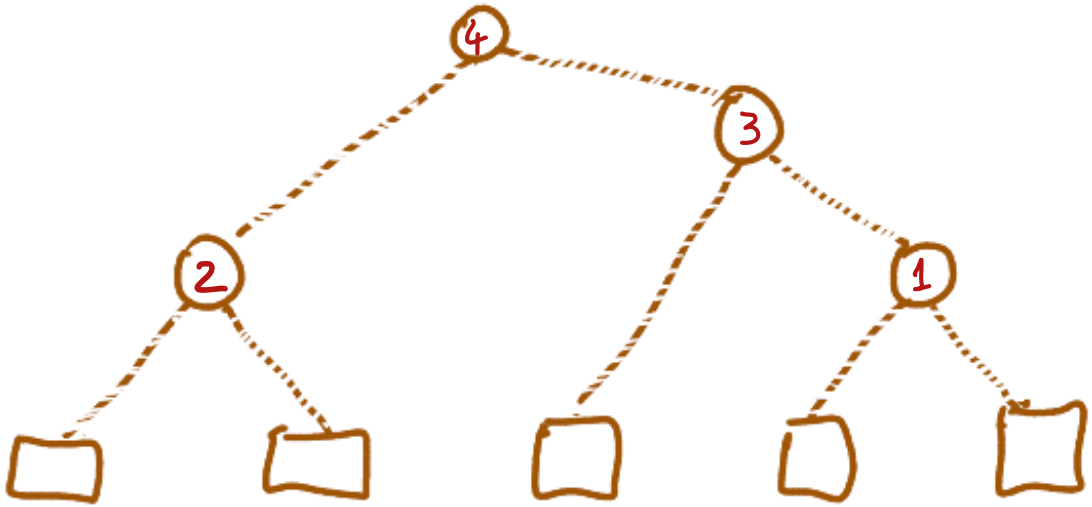
IN T_{cod}

n = 1

nodi interni = 0



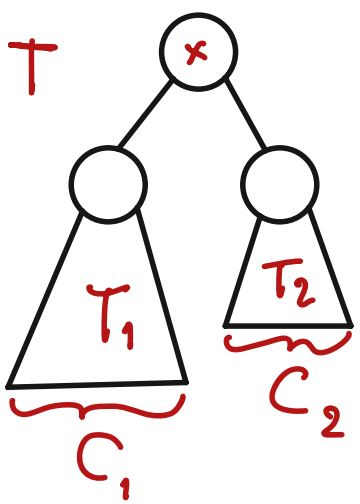
$= (m+p) - 1 = n - 1$



- PER COSTRUIRE UN ALBERO BINARIO PIENO CON n FOGLIE SI POSSONO EFFETTUARE $(n-1)$ OPERAZIONI DI MERGING (O FUSIONE)

COSTO DI UN'OPERAZIONE DI MERGING

DATO UN SISTEMA (C, f), SIANO $C_1, C_2 \subseteq C$ TALI CHE $C_1 \cap C_2 = \emptyset$ E SIANO T_1 E T_2 ALBERI DI DECODIFICA DI C_1 E C_2, RISPETT.



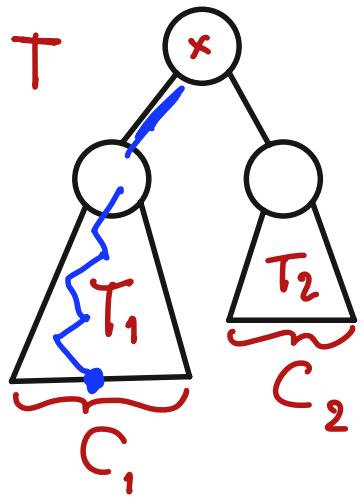
- L'ALBERO DI DECODIFICA T E' OTTENUTO DALLA FUSIONE DI T_1 E T_2 CON LA RADICE x

- INDICHIAMO L'OPERAZIONE DI MERGING DI T_1 E T_2 CON

  merging (T_1, T_2) OPPURE merging (x)

COSTO DELL' OPERAZIONE DI MERGING DI T_1 E T_2

$$B(\text{merging}(T_1,T_2)) := \sum_{c\in C_1} f(c) + \sum_{c\in C_2} f(c)$$



SI HA:

$$B(T) = \sum_{c\in C_1\cup C_2} f(c)\cdot d_T(c)$$

$$= \sum_{c\in C_1} f(c)\cdot d_T(c) + \sum_{c\in C_2} f(c)\cdot d_T(c)$$

$$= \sum_{c\in C_1} f(c)\cdot (d_{T_1}(c)+1) + \sum_{c\in C_2} f(c)\cdot (d_{T_2}(c)+1)$$

$$= \sum_{c\in C_1} f(c)\cdot d_{T_1}(c) + \sum_{c\in C_2} f(c)\cdot d_{T_2}(c) + \sum_{c\in C_1} f(c) + \sum_{c\in C_2} f(c)$$

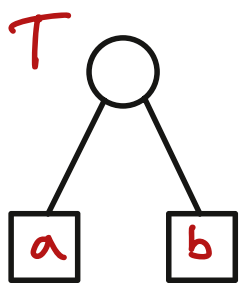
$$= B(T_1) + B(T_2) + B(\text{merging}(T_1,T_2))$$

LEMMA IL COSTO $B(T)$ DI UN ALBERO DI DECODIFICA T E' UGUALE ALLA SOMMA DEI COSTI DELLE OPERAZIONI DI MERGING NECESSARIE A COSTRUIRE T.

DIM. PER INDUZIONE SUL NUMERO m DI NODI INTERNI DI T.

PONIAMO: $int(T) :=$ INSIEME DEI NODI INTERNI DI T

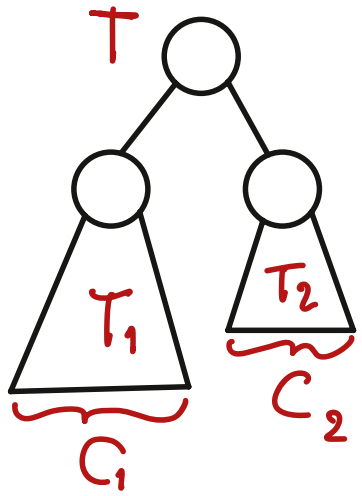
CASO BASE: $m = 1$



$$B(T) = f(a) + f(b) = B(merging(root(T)))$$

$$= \sum_{v \in int(T)} B(merging(v))$$

PASSO INDUTTIVO:



$$B(T) = B(T_1) + B(T_2) + B(\text{merging}(\text{root}(T)))$$

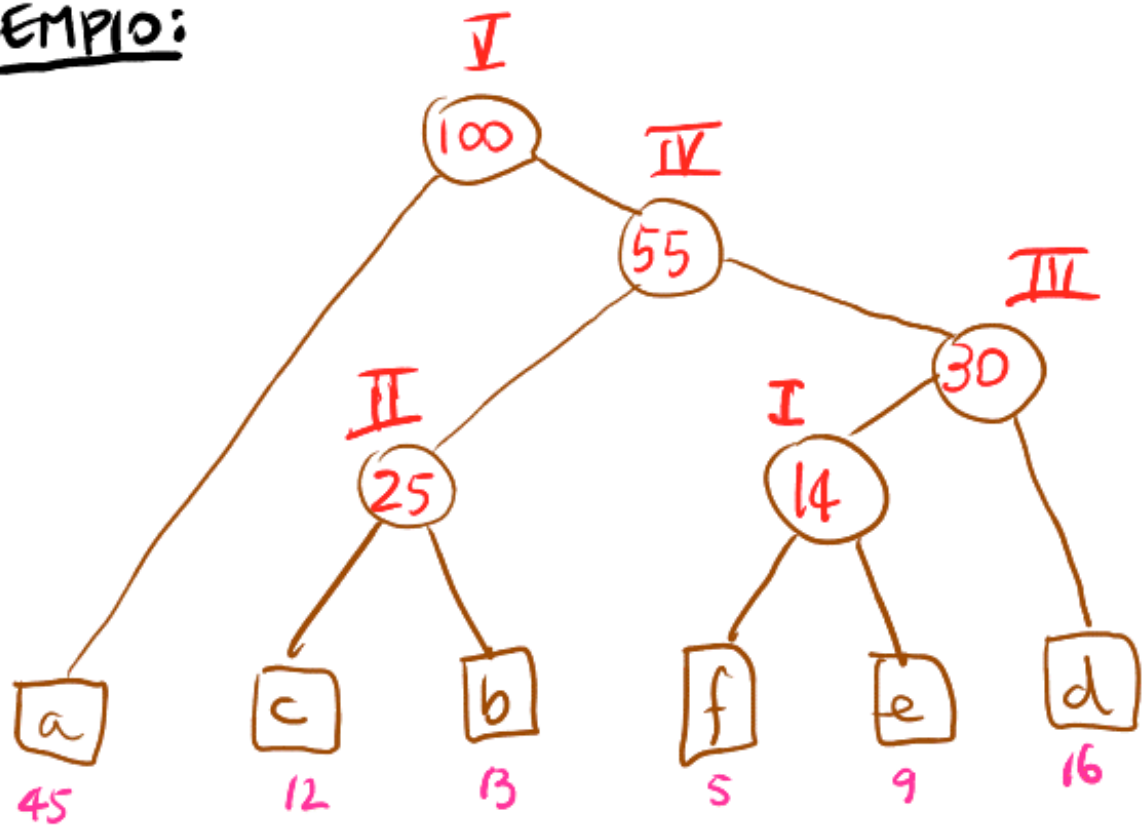
$$= \sum_{v \in \text{int}(T_1)} B(\text{merging}(v)) + \sum_{v \in \text{int}(T_2)} B(\text{merging}(v))$$

$$+ B(\text{merging}(\text{root}(T)))$$

$$= \sum_{v \in \text{int}(T)} B(\text{merging}(v))$$

■

ESEMPIO:



14 +
30 +
25 +
55 +
100

224

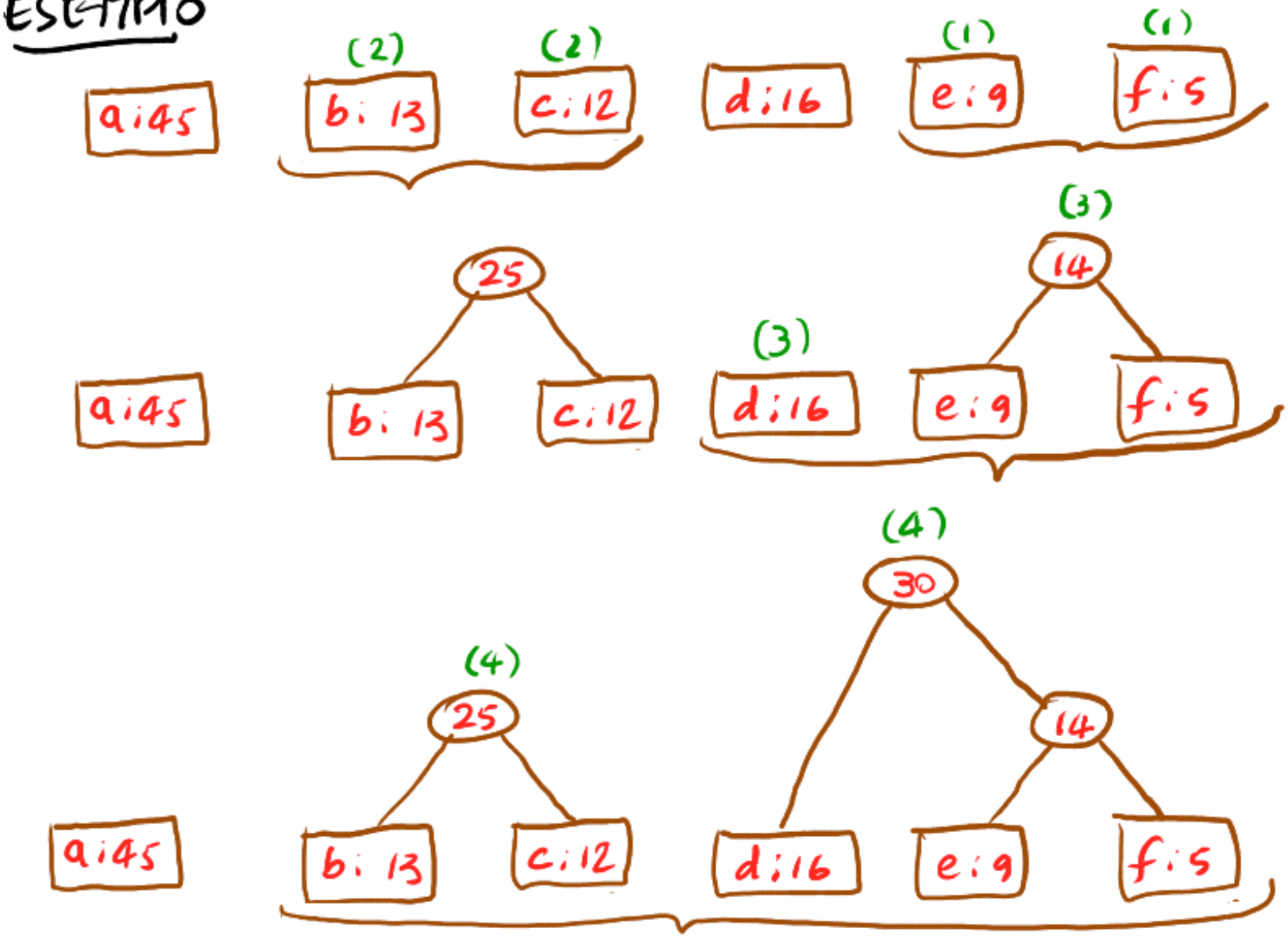
- UNA POSSIBILE STRATEGIA "GREEDY" PER COSTRUIRE UN ALBERO DI COSTO MINIMO CONSISTE NELL'EFFETTUARE LE OPERAZIONI DI MERGING DI COSTO MINIMO

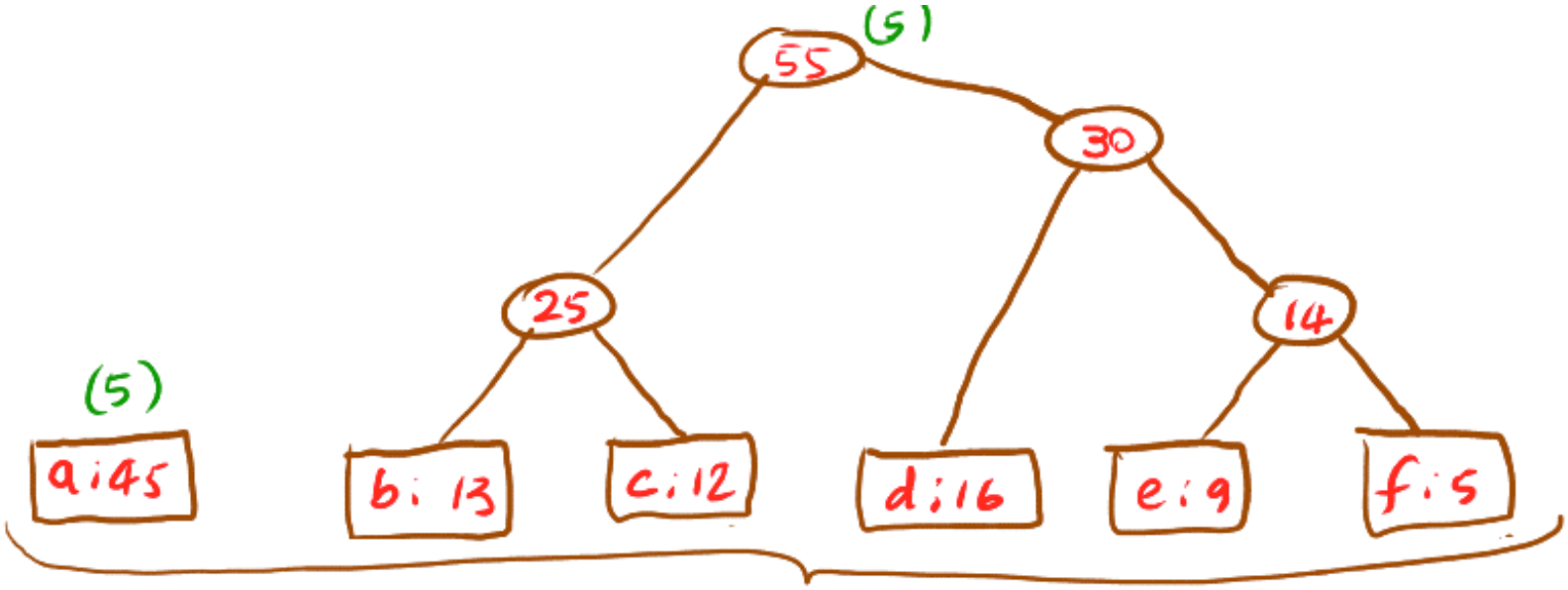
```
HUFFMAN(C, f)
 n := |C|
 Q := make_queue(C, f)
 for i := 1 to n-1 do
     - SI ALLOCHI UN NUOVO NODO INTERNO z
     left[z] := x := EXTRACT_MIN(Q)
     right[z] := y := EXTRACT_MIN(Q)
     f[z] := f[x] + f[y]
     INSERT(Q, z, f)
 return EXTRACT_MIN(Q)
```

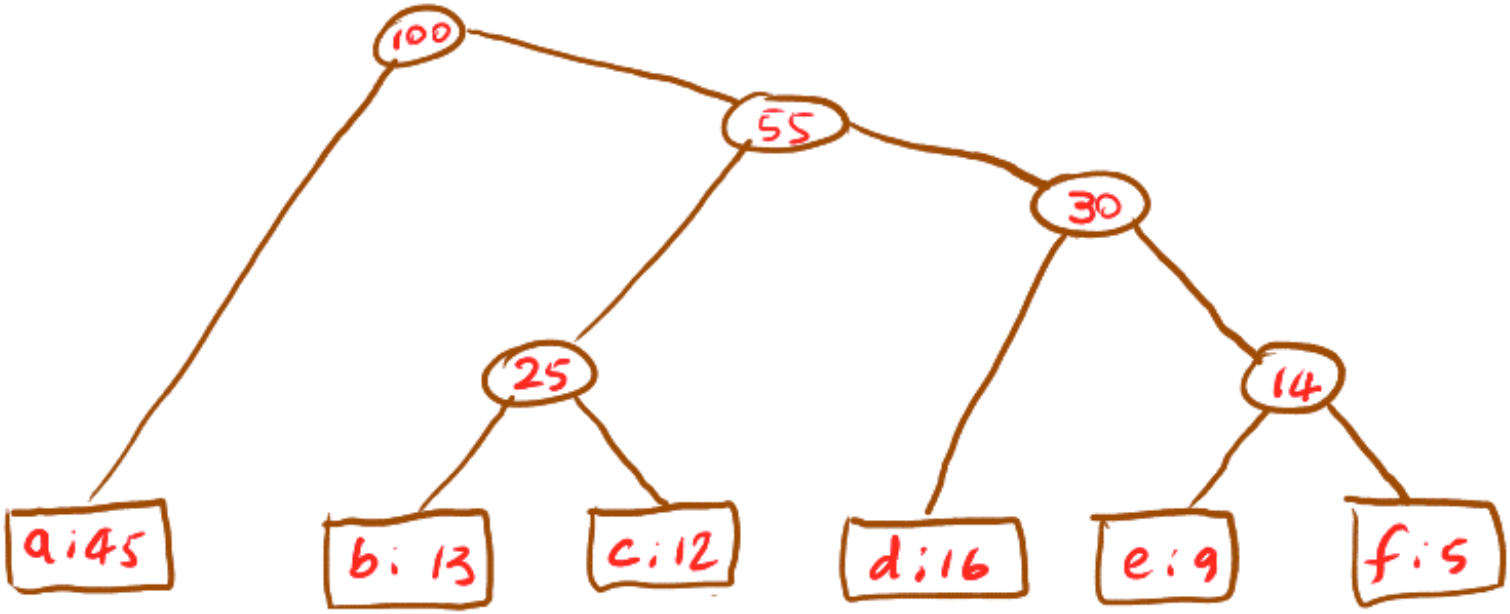
COMPLESSITA'

$(2n-1)$	EXTRACTMIN	$O(n \lg n)$
$(n-1)$	INSERT	$O(n \lg n)$
	BUILDHEAP	$O(n)$
		$O(n \lg n)$

ESEMPIO



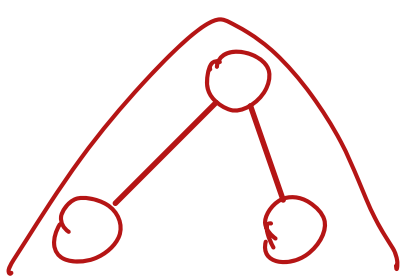

(5)
55
30
25
14
(5)
a:45
b:13
c:12
d:16
e:9
f:5

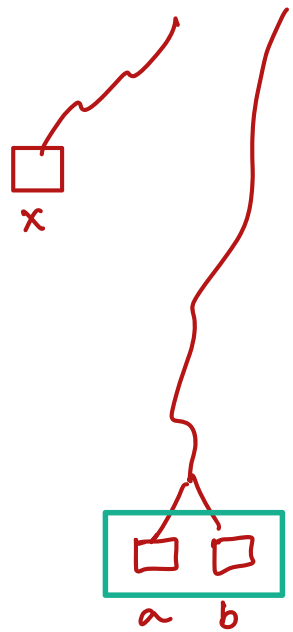

100
55
30
25
14
a:45
b:13
c:12
d:16
e:9
f:5

CORRETTEZZA DELL'ALGORITMO DI HUFFMAN

LEMMA (PROPRIETA' DI SCELTA GREEDY)

SIA C UN ALFABETO ED $f : C \to \mathbb{N}$ UNA FUNZIONE FREQUENZA.
SIANO x ED y I DUE CARATTERI IN C DI FREQUENZA MINIMA.
ALLORA ESISTE UN CODICE OTTIMO PREFISSO PER C IN CUI LE CODIFICHE DI x ED y DIFFERISCONO SOLO PER L'ULTIMO BIT.





scambio
$x \leftrightarrow a$
$y \leftrightarrow b$

$f(x) \le f(y)$

$f(a) \le f(b)$

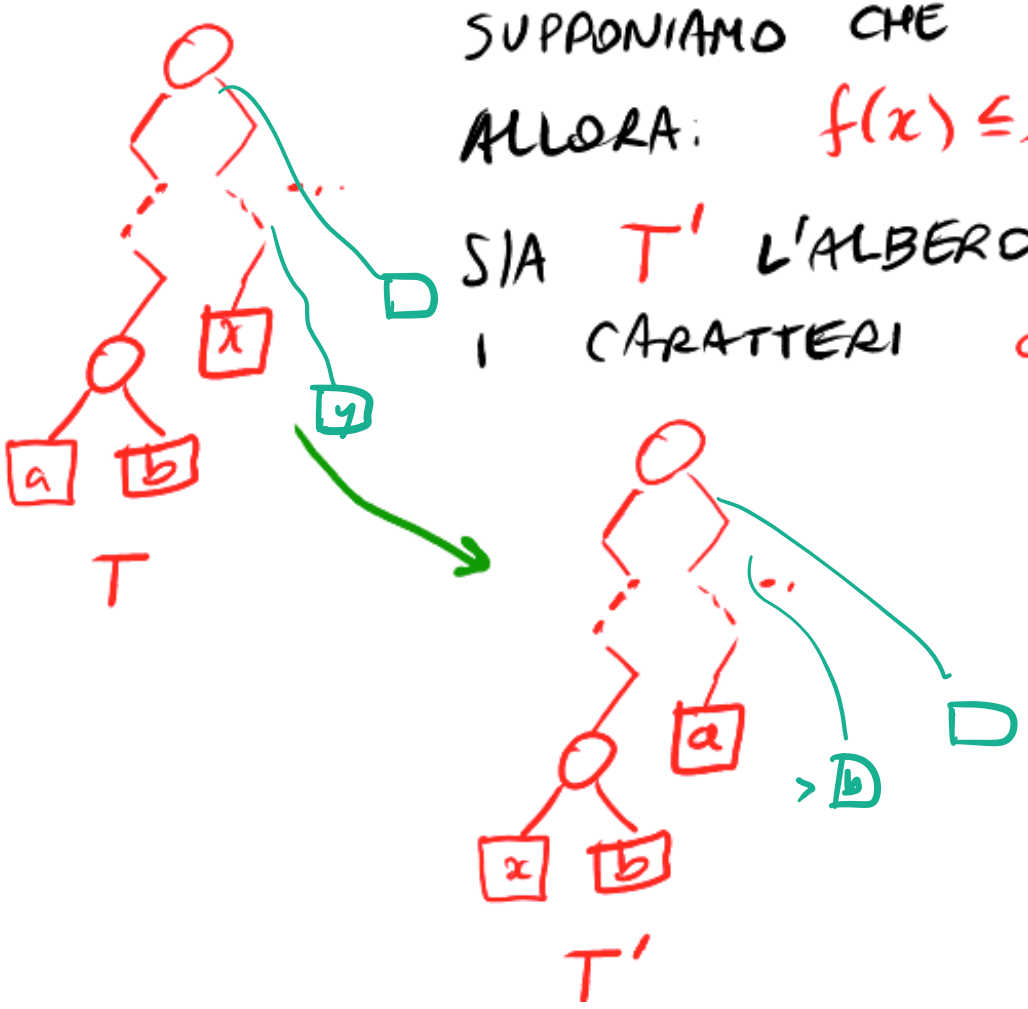
foglie a profondità massima

DIM. SIANO a E b DUE CARATTERI RESIDENTI SU FOGLIE SORELLE DI PROFONDITÀ MASSIMA IN UN ALBERO OTTIMO T.

SUPPONIAMO CHE $f(a) \leq f(b)$ E $f(x) \leq f(y)$.

ALLORA: $f(x) \leq f(a)$ E $f(y) \leq f(b)$,

SIA T' L'ALBERO OTTENUTO DA T SCAMBIANDO I CARATTERI a ED x,



SI HA:

$$B(T) = \sum_{c \in C} f(c)\, d_T(c)$$

$$= \sum_{c \in C \setminus \{a,x\}} f(c)\, d_T(c) + f(a)\, d_T(a) + f(x)\, d_T(x)$$

$$+ f(a)\, d_{T'}(a) + f(x)\, d_{T'}(x)$$

$$- f(a)\, d_{T'}(a) - f(x)\, d_{T'}(x) \Big] = 0$$

$$= \sum_{c \in C \setminus \{a,x\}} f(c)\, d_{T'}(c) + f(a)\, d_T(a) + f(x)\, d_T(x)$$

$$+ f(a)\, d_{T'}(a) + f(x)\, d_{T'}(x)$$

$$- f(a)\, d_{T'}(a) - f(x)\, d_{T'}(x)$$

$$= \sum_{c \in C} f(c)\, d_{T'}(c) + f(a)\, d_T(a) + f(x)\, d_T(x)$$

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

$$= B(T') + f(a)\big(d_T(a) - d_T(x)\big)$$

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

$$= B(T') + \big(f(a) - f(x)\big) \cdot \big(d_T(a) - d_T(x)\big)$$

CIOE'

$$B(T) = B(T') + \left(f(a) - f(x)\right) \cdot \left(d_T(a) - d_T(x)\right)$$

E PERTANTO:

$$B(T) - B(T') = \underbrace{\left(f(a) - f(x)\right)}_{\geq 0} \cdot \underbrace{\left(d_T(a) - d_T(x)\right)}_{\geq 0} \geq 0$$

(IN QUANTO $f(a) \geq f(x)$ E $d_T(a) \geq d_T(x)$)

DA CUI

$$B(T) \geq B(T').$$

- SIA T'' L'ALBERO OTTENUTO DA T' SCAMBIANDO I CARATTERI b ED y.
- ANALOGAMENTE A QUANTO VISTO PRIMA, SI HA:

  $B(T') \geq B(T'')$ (IN QUANTO $f(b) \geq f(y)$ E $d_{T'}(b) \geq d_{T'}(y)$)
- PERTANTO: $B(T) \geq B(T'')$

- POICHE' T E' OTTIMO, $B(T'') \geq B(T)$, E QUINDI $B(T'')$ E' ANCH'ESSO OTTIMO
- INOLTRE IN T'' I CARATTERI x E y RISIEDONO SU FOGLIE SORELLE E QUINDI I LORO CODICI DIFFERISCONO SOLO PER L'ULTIMO BIT. ■

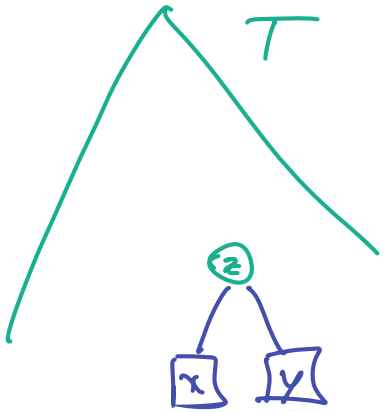
LEMMA (PROPRIETA' DELLA SOTTOSTRUTTURA OTTIMA)

- SIA C UN ALFABETO ED $f : C \to \mathbb{N}$ UNA FUNZIONE FREQUENZA.
- SIANO x ED y I DUE CARATTERI IN C DI FREQUENZA MINIMA.
- SIA $C' = (C \setminus \{x, y\}) \cup \{z\}$, CON $z \notin C$.
- SIA $f' : C' \to \mathbb{N}$ TALE CHE: $f'(c) = \begin{cases} f(c) & \text{SE } c \neq z \\ f(x) + f(y) & \text{SE } c = z \end{cases}$
- SIA T' UN ALBERO OTTIMO PER (C', f').
- SIA T L'ALBERO OTTENUTO DA T' SOSTITUENDO LA FOGLIA z CON UN NODO INTERNO AVENTE COME FIGLI DUE FOGLIE ETICHETTATE CON x ED y, RISPETTIVAMENTE.

ALLORA T E' OTTIMO PER (C, f).

DIMOSTRAZIONE

SI HA:



$$B(T) = \sum_{c \in C} f(c) \cdot d_T(c)$$

$$= \sum_{c \in C \setminus \{x,y\}} f(c) \cdot d_T(c) + f(x) \cdot d_T(x) + f(y) \cdot d_T(y)$$

$$= \sum_{c \in C \setminus \{x,y\}} f'(c) \cdot d_{T'}(c) + f(x) \cdot (d_{T'}(z) + 1) + f(y) \cdot (d_{T'}(z) + 1)$$

$$= \sum_{c \in C \setminus \{x,y\}} f'(c) \cdot d_{T'}(c) + \big(f(x) + f(y)\big) \cdot d_{T'}(z) + \big(f(x) + f(y)\big)$$

$$= \sum_{c \in C \setminus \{x,y\}} f'(c) \cdot d_{T'}(c) + f'(z) \cdot d_{T'}(z) + \big(f(x) + f(y)\big)$$

$$= \sum_{c \in C'} f'(c) \cdot d_{T'}(c) + \big(f(x) + f(y)\big)$$

$$= B(T') + \big(f(x) + f(y)\big)$$

PERTANTO: $B(T') = B(T) - (f(x) + f(y))$

- SE T NON FOSSE OTTIMO PER (C,f), ESISTEREBBE UN ALBERO T'' OTTIMO PER (C,f) TALE CHE: $B(T'') < B(T)$.
- GRAZIE AL LEMMA PRECEDENTE, POSSIAMO SUPPORRE CHE x E y SI TROVINO SU FOGLIE SORELLE IN T''.
- SIA T''' OTTENUTO DA T'', SOSTITUENDO IL PADRE DI x E y CON UNA FOGLIA z CON FREQUENZA $f(x)+f(y)$.
- ALLORA:

$$\begin{aligned} B(T''') &= B(T'') - f(x) - f(y) \\ &< B(T) - f(x) - f(y) \\ &= B(T') \end{aligned}$$

CONTRADDICENDO L'OTTIMALITA' DI T' PER (C',f').

- PERTANTO T E' OTTIMO PER (C,f). ∎

"SI␣SENTIVANO␣CONTINUI␣CORI␣DI␣
PO PO PO PO PO PO PO PO PO PO PO PO PO PO

- S → 2
- I → 6
- ␣ → 5
- E → 1
- N → 4
- T → 2
- V → 1
- A → 1
- O → 17
- C → 2
- U → 1
- R → 1
- D → 1
- P → 14

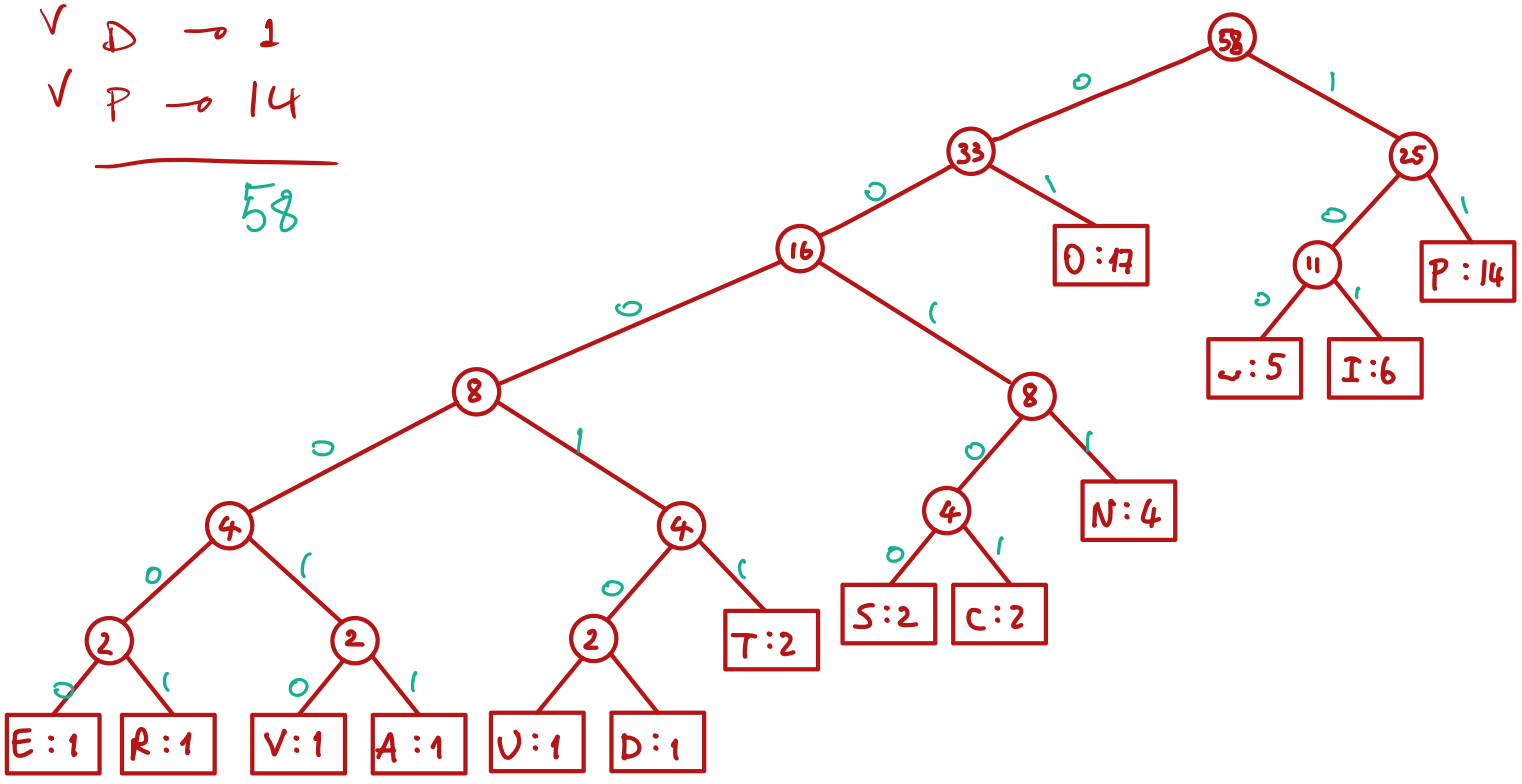
(10) (10) (20) (18)

58

4 bit x carattere

4 bit x 58 = 232 bit

in una codifica a lungh. fissa



$6 \times 6 = 36$

$6 \times 5 = 30$

$4 \times 4 = 16$

$11 \times 3 = 33$

$31 \times 2 = 62$

177 bit

$232 - 177 = 55$ bit risparmiati su 232

$\frac{55}{232} \cdot 100 \simeq 23.71\ \%$

RIASSUMENDO:

LUNGHEZZA FISSA A 4 BIT	232 bit
CODICE PREFISSO OTTIMO	177 bit
RISPARMIO	55 bit
RISPARMIO PERCENTUALE	~ 23,71 %