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.	COD1 (8 BIT)	COD2 (3 bit)	COD3	
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

$\underbrace{\text{800 bit} \quad \text{300 bit}}_{\text{LUNGHEZZA FISSA}}$ $\underbrace{\qquad\qquad}_{\text{LUNGH. VARIABILE}}$

25% IN MENO

ES. a b a c

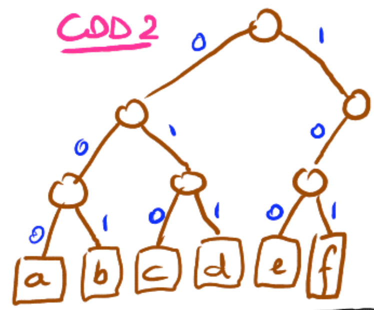
COD 2
$\underline{000}\,\underline{001}\,\underline{000}\,\underline{010}$

COD 3
$\underline{0}\,\underline{101}\,\underline{0}\,\underline{100}$

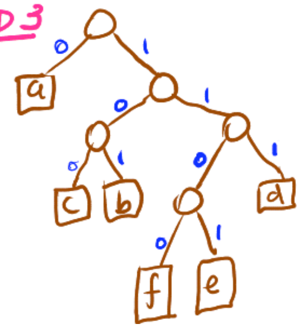
ALBERI DI DECODIFICA

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

$\underbrace{000}_{a}\underbrace{001}_{b}\underbrace{000}_{a}\underbrace{010}_{c}$

COD 3

$\underbrace{0}_{a}\underbrace{101}_{b}\underbrace{0}_{a}\underbrace{100}_{c}$

- 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

$\underbrace{01}_{b}\underbrace{11}_{c}\underbrace{11}_{c}\underbrace{11}_{c}\underbrace{11}_{c}\underbrace{11}_{c}$

ESEMPIO DI CODICE NON PREFISSO AMBIGUO

a	0
b	1
c	01

01

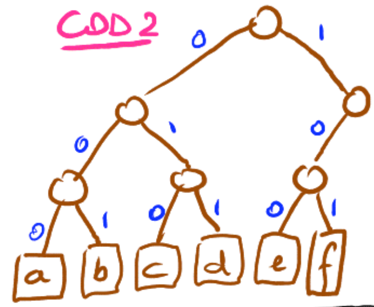
ab c

?

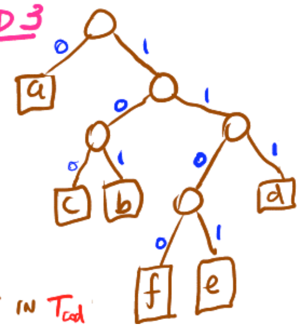
ALBERI DI DECODIFICA

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



COMPLESSITA' 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

C : ALFABETO

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

$d_{T_{cod}}$: PROFONDITA' IN T_{cod}

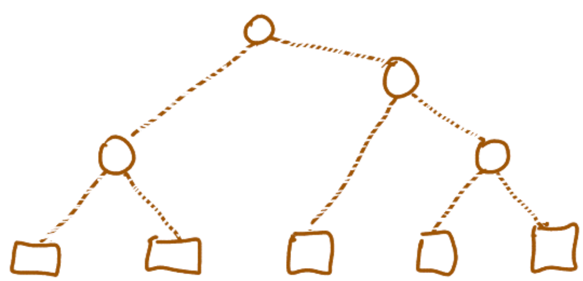
PROBLEMA: TRA TUTTI GLI ALBERI DI DECODIFICA RELATIVI AD UN SISTEMA (C, f) (DOVE $f : C \to \mathbb{N}$) DETERMINARE QUELLO DI COSTO MINIMO, CIOE' L'ALBERO BINARIO DI DECODIFICA T TALE CHE

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

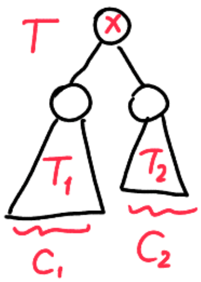
SIA MINIMO

OSSERVAZIONE: POSSIAMO LIMITARE LA NOSTRA RICERCA AGLI ALBERI BINARI PIENI, QUELLI CIOE' PRIVI DI NODI INTERNI CON MENO DI DUE FIGLI.

OSSERVAZIONE: IL NUMERO DI NODI INTERNI IN UN ALBERO BINARIO PIENO CON n FOGLIE E' $n-1$.



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



$C_1 \cup C_2 = C$

$c \in C_1 \quad d_{T_1}(c)+1 = d_T(c)$

$c \in C_2 \quad d_{T_2}(c)+1 = d_T(c)$

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

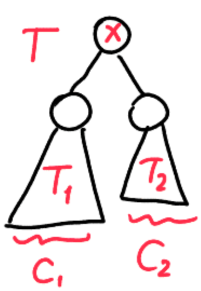
$$= \sum_{c \in C_1} f(c) \cdot d_T(c) + \sum_{c \in C_2} f(c) d_T(c)$$

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

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

$$+ \sum_{c \in C_2} f(c) d_{T_2}(c) + \sum_{c \in C_2} f(c)$$

$$= B(T_1) + B(T_2) + \sum_{c \in C} f(c)$$



$C_1 \cup C_2 = C$

$c \in C_1 \quad d_{T_1}(c)+1 = d_T(c)$

$c \in C_2 \quad d_{T_2}(c)+1 = d_T(c)$

$$B(T) = B(T_1) + B(T_2) + \sum_{c \in C} f(c)$$

$$\Delta B = B(T) - (B(T_1) + B(T_2))$$

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

← COSTO DELL' OPERAZIONE DI MERGING DI T_1 E T_2

14
30
25
55
100

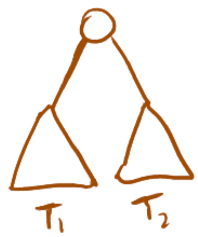
224

PER INDUZIONE SULL'ALTEZZA DI T, SI DIMOSTRA CHE:

$B(T)$ = SOMMA DEI COSTI DI TUTTE LE OPERAZIONI DI MERGING

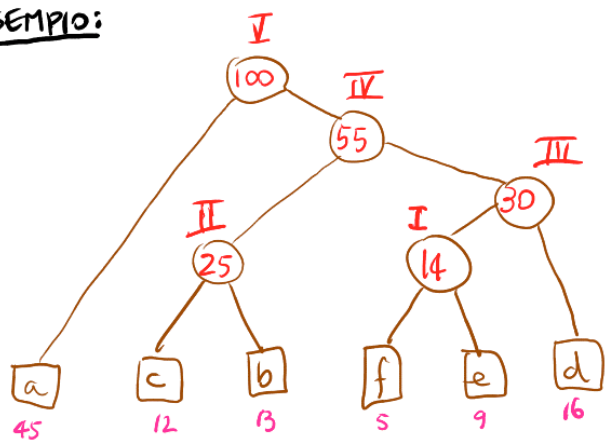
CASO BASE: $height(T) = 0 \quad B(T) = 0$

PASSO INDUTTIVO:



$$\begin{aligned} B(T) &= B(T_1) + B(T_2) + \sum_{c \in C} f(c) \\ &= \sum merging(T_1) \\ &\quad + \sum merging(T_2) \\ &\quad + costo\_merging(T_1, T_2) \\ &= \sum merging(T) \end{aligned}$$

ESEMPIO:



$$\begin{array}{r} 14 + \\ 30 + \\ 25 + \\ 55 + \\ 100 \\ \hline 224 \end{array}$$

- 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'

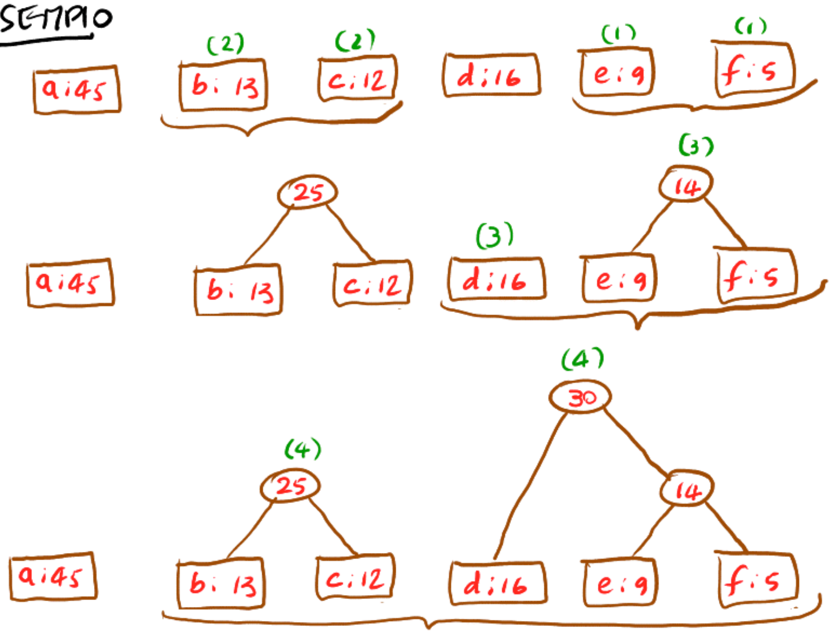
$(2n-1)$ EXTRACT_MIN $O(n \lg n)$

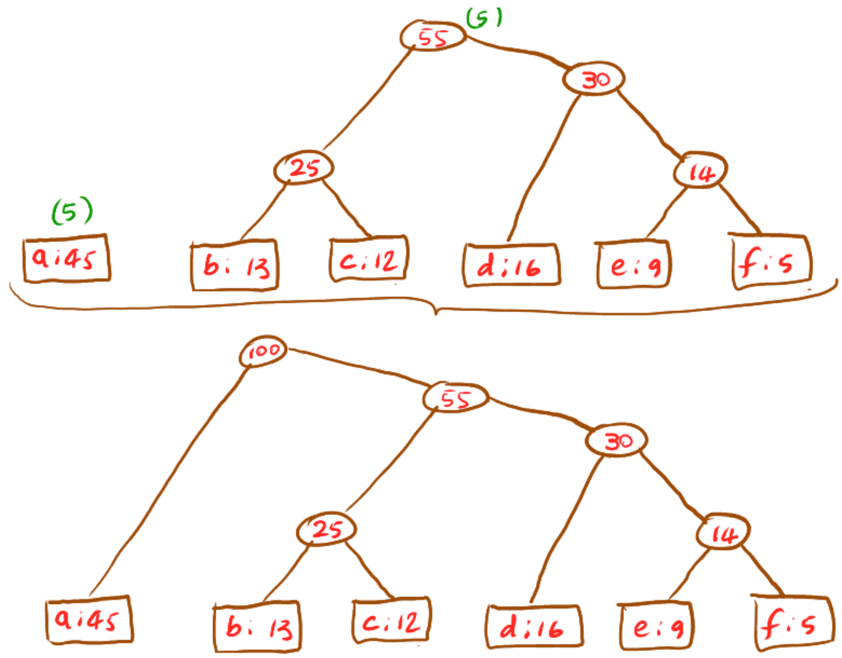
$(n-1)$ INSERT $O(n \lg n)$

BUILDHEAP $O(n)$

$O(n \lg n)$

ESEMPIO

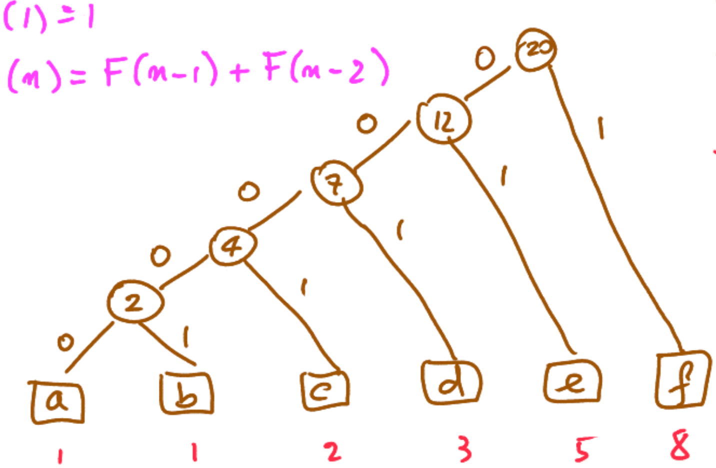




LOWER-BOUND SULLA COMPLESSITA': $\Omega(n \lg n)$

$$\begin{cases} F(0) = 1 \\ F(1) = 1 \\ F(n) = F(n-1) + F(n-2) \end{cases}$$

a	00000
b	00001
c	0001
d	001
e	01
f	1



CORRETTEZZA DELL'ALGORITMO DI HUFFMAN

LEMMA SIA C UN ALFABETO E $f: C \to \mathbb{N}$ 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.

DIM. SIANO a E b DUE CARATTERI RESIDENTI SU FOGLIE SORELLE DI PROFONDITA' 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:

$$B(T) - B(T') = \sum_{c \in C} f(c)\, d_T(c) - \sum_{c \in C} f(c)\, d_{T'}(c)$$

$$= f(a) d_T(a) + f(x) d_T(x) - f(a) d_{T'}(a) - f(x) d_{T'}(x)$$

$$= f(a) d_T(a) + f(x) d_T(x) - f(a) d_T(x) - f(x) d_T(a)$$

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

$$= \big(f(a) - f(x)\big)\big(d_T(a) - d_T(x)\big) \geq 0$$

POICHE':

- $c \in C \setminus \{a, x\} \longrightarrow d_T(c) = d_{T'}(c)$
- $d_{T'}(a) = d_T(x)$
- $d_{T'}(x) = d_T(a)$

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

  $B(T') - B(T'') \geq 0$
- PERTANTO: $B(T) - B(T'') \geq 0$, DA CUI $B(T) \geq B(T'')$
- 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

- SIA C UN ALFABETO E $f : C \to \mathbb{N}$ 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).

DIM. SI HA:

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

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

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

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

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

DA CUI: $B(T') = B(T) - f(x) - f(y)$

- 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). ∎