



# Codici di Huffman

Paolo Costa

[paolo.costa@polimi.it](mailto:paolo.costa@polimi.it)



# Alberi: codifica di Huffman

- ✘ Codici ASCII: 8 bit per carattere - codifica a lunghezza fissa
- ✘ E' possibile sfruttare la freq. relativa delle lettere per risparmiare spazio con una codifica a lunghezza variabile



# Frequenza relativa (inglese)

$f$  e` la frequenza relativa attesa per 1000 lettere, calcolata in un insieme campione di documenti - consideriamo per semplicità questo insieme parziale di lettere:

$$f(Z)=2, f(K)=7, f(F)=24,$$

$$f(C)=32, f(U)=37, f(D)=42,$$

$$f(L)=42, f(E)=120$$



# Costruzione dell'albero di Huffman

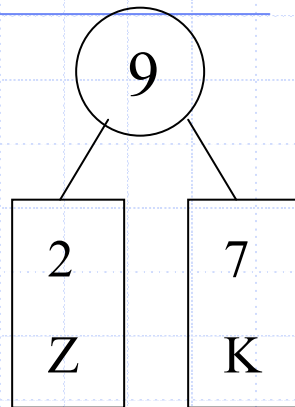
il campione selezionato verrà usato per costruire un albero di codifica di Huffman, dipendente dalla freq.

**Passo 1: lista ordinata per freq.**

(Z, 2), (K, 7), (F, 24), (C, 32), (U, 37),  
(D, 42), (L, 42), (E, 120)



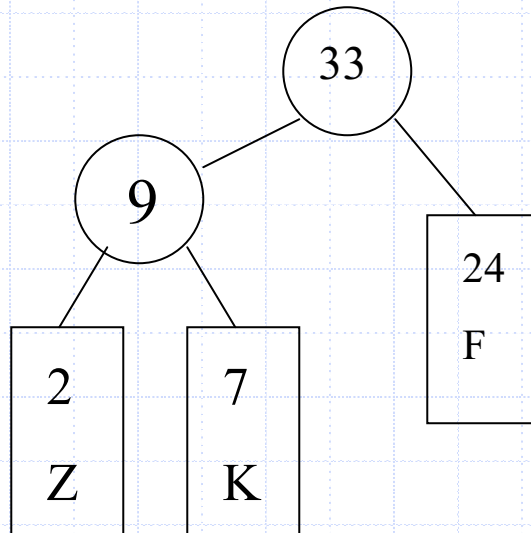
Passo 2:



(F, 24), (C, 32), (U, 37), (D, 42), (L, 42), (E, 120)

Passo 3:

(C, 32),



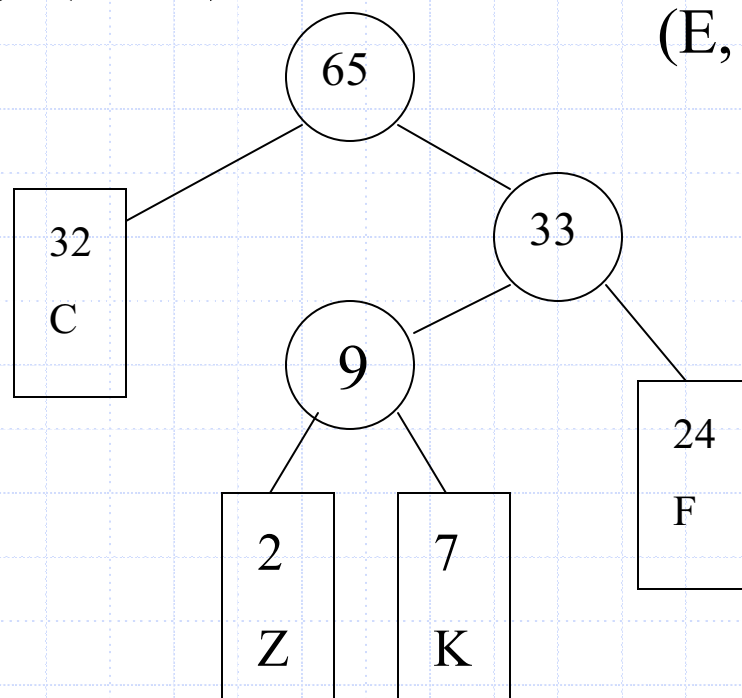
(U, 37), (D, 42), (L, 42), (E, 120)



Passo 4:

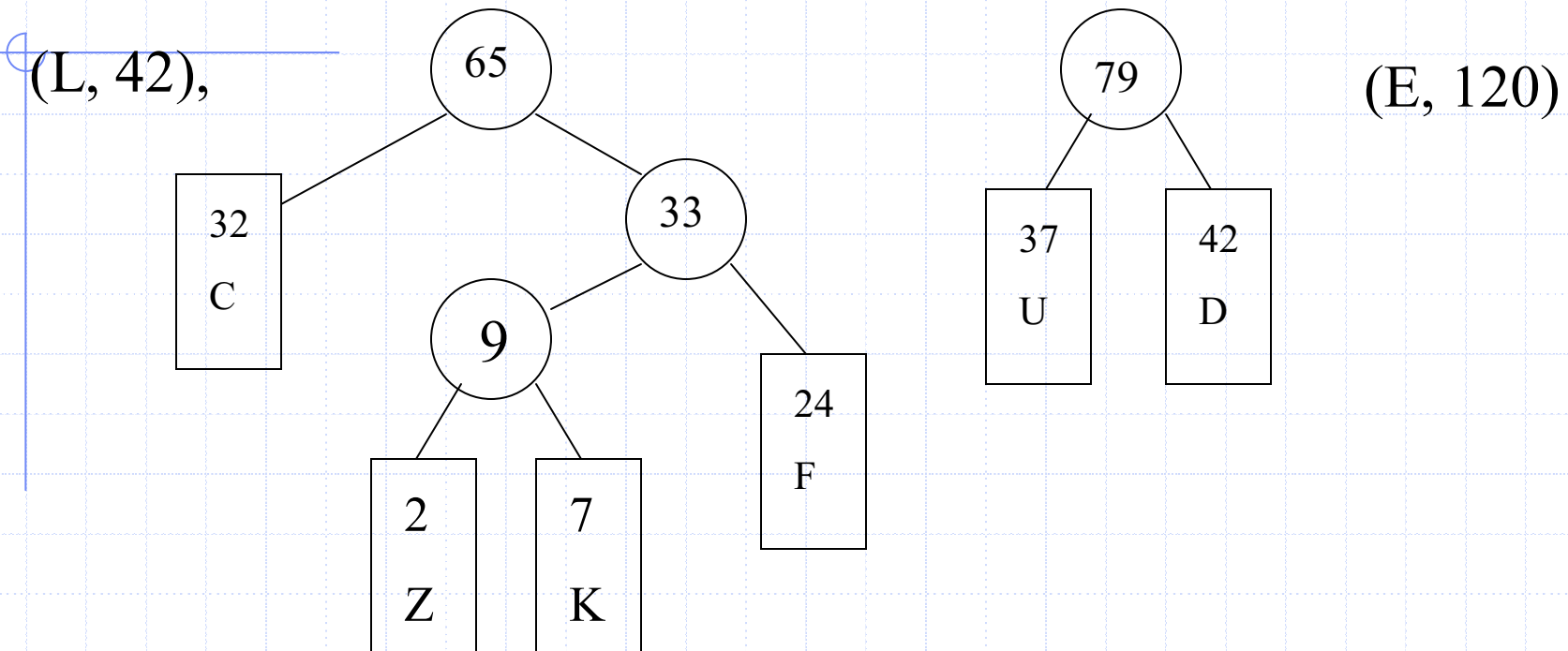
(U, 37), (D, 42), (L, 42),

(E, 120)



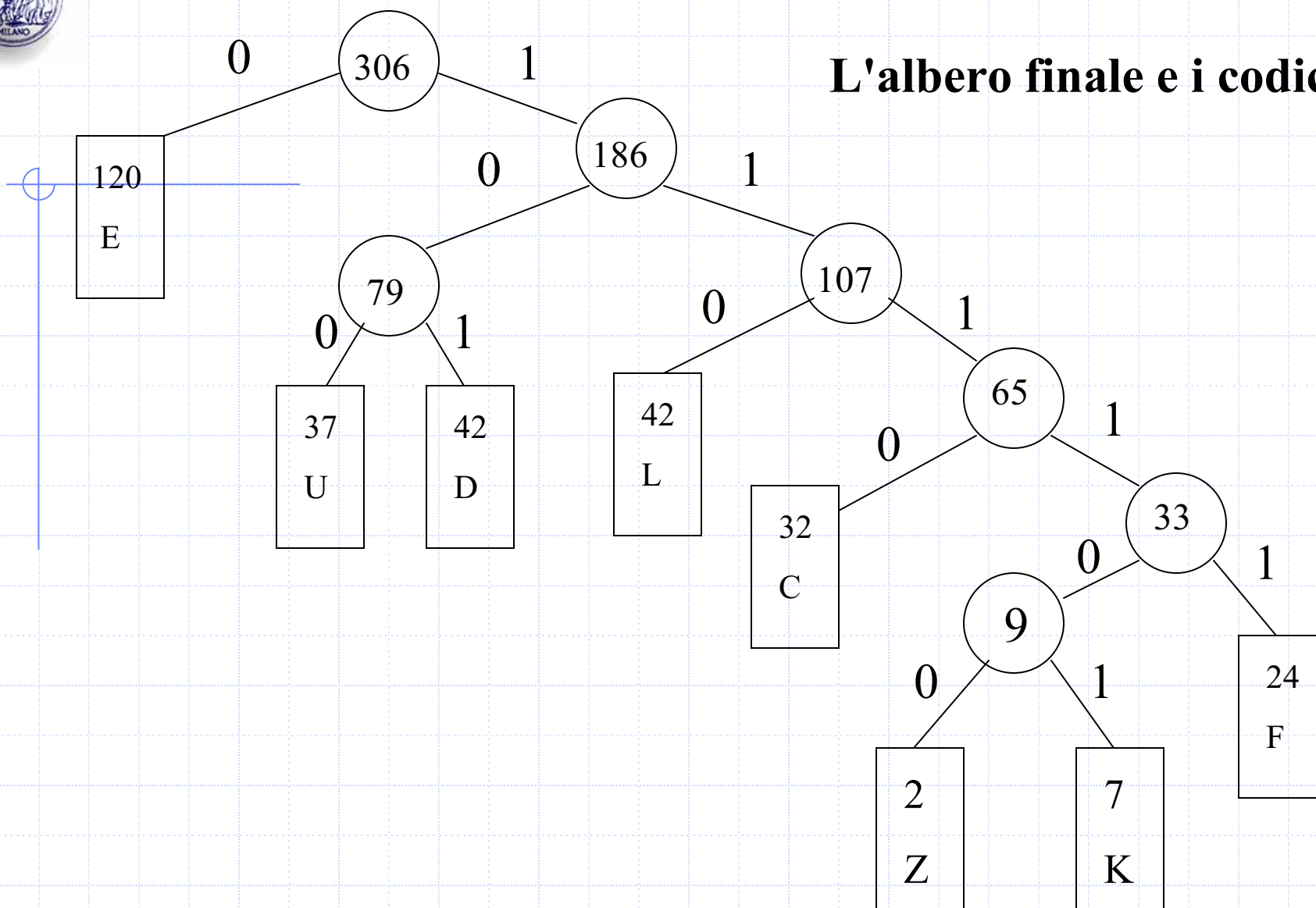


# Passo 5:





## L'albero finale e i codici







Dunque i codici sono:

E: 0

U: 100

D: 101

L: 110

C: 1110

F: 11111

Z: 111100

K: 111101

Nota: godono della *proprietà di prefisso*,  
cioè nessun codice è prefisso d'un altro

qualche esempio:

DEED = 101 0 0 101

Decodifica di

1011001110111101

(duck)



# Risparmio ?!?

- ✘ Per questo insieme di frequenze il costo totale di compressione diventa:

$$1 \times 120 + 3 \times 121 + 4 \times 32 + 5 \times 24 + 6 \times 9 = 785$$

- ✘ contro un costo con una codifica standard di:

$$3 \times (120 + 121 + 32 + 24 + 9) = 3 \times 306 = 918$$