

Huffman

È un algoritmo di codifica che usa i codici prefissi. Invece di usare codifiche fisse si usano codifiche a lunghezza variabile cioè l'albero di Huffman dove ogni foglio corrisponde a una lettera ed ad ognuno di queste viene associato un codice univoco o prefisso. L'associazione ovviamente non è casuale se un carattere ha frequenza maggiore nel testo questo avrà un numero di bit inferiore, questo per aumentare la compressione.

La funzione $v(T)$ valuta l'albero

$$v(T) = \sum_{c \in \Sigma} f(c) \cdot l(c)$$

\nearrow frequente
 \nwarrow lunghezza codice
 \nwarrow carattere dell'alfabeto

L'algoritmo per la creazione dell'albero deve minimizzare questo valore

L'albero risultante dovrebbe avere $|C|$ foglie e $|C|-1$ nodi interni dove $C = \text{alfabeto}$
 \rightarrow rappresenta anche il numero di fusioni

Huffman usa una scelta globale per la creazione dell'albero con tempo $O(n \log n)$

Miglior di Huffman non si può fare

HUFFMAN (F, n)
 $Q \leftarrow \Sigma$ \nearrow fusioni
 \nwarrow Heap che contiene l'alfabeto
 $n = \text{numero di caratteri in } \Sigma$
for $i \leftarrow 1$ to $n-1$ do

```

FOR  $x \leftarrow 1$  TO  $(n-1)$  DO
   $x \leftarrow \text{EXTRACT-MIN}(Q)$ 
   $y \leftarrow \text{EXTRACT-MIN}(Q)$ 
   $z \leftarrow \text{NEW NODE}$ 
   $F(z) \leftarrow F(y) + F(x)$ 
   $z \leftarrow \text{LEFT}(x)$ 
   $z \leftarrow \text{RIGHT}(y)$ 
   $Q \leftarrow \text{INSERT}(z)$ 
 $T \leftarrow \text{EXTRACT-MIN}(Q)$ 
RETURN  $T$ 

```

} perde con frequenza minore

viene tornata la radice