

A B C D D A A E A D  
C C A D A D A B G B  
D A A B A B E D A A  
A B A A A A E A D  
B B A B A C B A B D  
A A E A C A A C F D  
E B B B A B C A B G  
A B A B A C A F G B  
A C B A B A F A F H  
B B A C B B A F A H

Figura 5A - Immaginiamo di voler trasferire questo insieme di caratteri.

esempio che alcuni caratteri siano più frequenti e altri meno frequenti. Fortunatamente ciò accade anche nella lingua italiana, dove consonanti come la Q sono certamente più rare della S o della T o roba simile.

Oltre a questo, per poter funzionare, il codice Huffman necessita di trasferimento seriale che, come detto prima,

invia un bit dopo l'altro. A questo punto passiamo alla figura 5A e come dice la didascalia immaginiamo di voler trasferire tale insieme di simboli. Da una veloce scorsa si nota subito una forte maggioranza di caratteri come la A e la B, contro un numero limitato di G e di H. Siamo in un caso abbastanza ottimale per applicare il codice Huffman. Senza di questo, come detto prima, dovremmo associare biunivocamente una sequenza di 0 e 1 ad ogni simbolo, per esempio così:

A 000  
B 001  
C 010  
D 011  
E 100  
F 101  
G 110  
H 111

E trasferendo l'insieme dei 100 simboli di 5A invieremmo in tutto 300 bit (100 simboli × 3 bit l'uno). Vediamo di fare qualcosa di meglio: seguite

sempre in figura 5. Il primo passo consiste nell'elencare tutti i simboli del testo ponendo sotto di essi la relativa occorrenza (fig. 5B).

Abbiamo infatti 40 A, 25 B, 10 C ecc.

Si tratta di costruire l'albero di figura 5F procedendo nel seguente modo: (fig. 5C) si fondono le occorrenze più piccole, 3 e 2 rispettivamente dei caratteri G e H ottenendo così una nuova occorrenza (5) dei caratteri G e H considerati insieme. Si itera il procedimento scegliendo sempre le occorrenze più piccole, considerando anche le occorrenze ottenute per fusione. Ad esempio, dopo le fusioni di fig. 5E le occorrenze da scegliere saranno 5 e 10 ottenendo così l'occorrenza 15.

Alla fine otterremo l'occorrenza 100 che è naturalmente quella dei simboli A...H, che vuol dire appunto che i simboli dalla A alla H sono in tutto presenti 100 volte, ovvio.

Ultimo passo, etichettare con un 1 tutti i rami sinistri dell'albero e con uno 0 tutti i rami destri come indicato in fig. 5F. Possiamo ora generare il codice Huffman di quei caratteri: partendo dalla occorrenza massima, raggiungiamo i vari caratteri prendendo nota del percorso che facciamo ossia prelevando gli 1 o gli 0 che incontriamo. Quindi alla lettera A associeremo il codice 1, al B il codice 01 (infatti per risalire da 100 fino alla B, occhio alla figura, dobbiamo percorrere prima un ramo destro poi un ramo sinistro). Tutto il codice sarà dunque:

	A	1	Tot.	1 bit
B	01	"	2	"
C	0011	"	4	"
D	0010	"	4	"
E	00011	"	5	"
F	00010	"	5	"
G	00001	"	5	"
H	00000	"	5	"

Se proviamo a trasferire il testo ci accorgeremo di aver risparmiato: infatti noi dobbiamo trasferire:

40 A =	40 × 1 =	40 bit +
25 B =	25 × 2 =	50 bit +
10 C =	10 × 4 =	40 bit +
10 D =	10 × 4 =	40 bit +
5 E =	5 × 5 =	25 bit +
5 F =	5 × 5 =	25 bit +
3 G =	3 × 5 =	15 bit +
2 H =	2 × 5 =	10 bit =
<b>totale = 245 bit</b>		

Come dire che ne abbiamo risparmiato ben 55 pari a più del 18%.

Per quel che riguarda la decodifica, ossia il ritornare ai caratteri man mano che si ricevono i bit serialmente, notiamo che anche questo è molto semplice, infatti come per i numeri te-

A B C D E F G H  
40 25 10 10 5 5 3 2

Figura 5B - Primo passo: si elencano i vari caratteri ponendo sotto di essi la relativa occorrenza nel testo da trasferire.

A B C D E F G H  
40 25 10 10 5 5 3 2  
5

Figura 5C - Si fondono insieme le occorrenze più piccole (3 e 2) ottenendo una nuova occorrenza (5) di G e di H (insieme).

A B C D E F G H  
40 25 10 10 5 5 3 2  
10 5

A B C D E F G H  
40 25 10 10 5 5 3 2  
20 10 5

Figura 5D e 5E - Si itera il procedimento prendendo sempre le occorrenze più piccole.