

Fondamenti di Informatica
per la Sicurezza
a.a. 2007/08

Esercizi di codifica binaria

Stefano Ferrari



UNIVERSITÀ DEGLI STUDI DI MILANO
DIPARTIMENTO DI TECNOLOGIE DELL'INFORMAZIONE

Stefano Ferrari ★ Università degli Studi di Milano

Fondamenti di Informatica per la Sicurezza ◇ Esercizi di codifica binaria ◇ a.a. 2007/08 - p. 1/20

Contare con le dita

D Fino a quanto si può contare usando solo le dita di due mani?

R Ipotizzando che ogni dito possa assumere solo due posizioni (disteso o chiuso):

- si hanno 10 elementi indipendenti;
- ogni elemento può assumere due configurazioni;
- il numero di configurazioni totali è $2^{10} = 1024$.

Si può quindi contare da 1 a 1024 (o da 0 a 1023!).

Stefano Ferrari ★ Università degli Studi di Milano

Fondamenti di Informatica per la Sicurezza ◇ Esercizi di codifica binaria ◇ a.a. 2007/08 - p. 2/20

Colori

D Quanti colori possono essere rappresentati in una codifica RGB a 1 byte per canale?

R Ognuno dei canali (R, G e B) ha a disposizione 8 bit.

In totale, dunque, $8 \times 3 = 24$ bit.

Sono quindi rappresentabili $2^{24} = 16\,777\,216$ colori.

La codifica a 24 bit è chiamata comunemente "a 16 milioni di colori".

Dimensione di una immagine digitale

D Quanti bit servono per rappresentare un'immagine 1024×768 a colori, 8 bit per canale colore?

R I pixel sono $1024 \times 768 = 786\,432$.

Ogni pixel è rappresentato da 3 colori (rosso, verde e blu) e ogni colore occupa 8 bit.

Una immagine così fatta, occupa quindi:
 $786\,432 \times 3 \times 8 = 18\,874\,368$ bit.

Essi equivalgono a $18\,874\,368/8 = 2\,359\,296$ byte.

Cioè $2\,359\,296/1024 = 2304$ kiB (*kibibyte*).

O $2304/1024 = 2,25$ MiB (*mebibyte*).

Codifica per le carte da briscola

- D** Quale può essere una buona codifica per le carte di un mazzo da briscola?
- R** Poiché il mazzo di carte da briscola è composto da 40 carte, si dovranno utilizzare 6 bit:
- $\lceil \log_2 40 \rceil = 6$
- Uno dei modi di ottenere una codifica è individuare una enumerazione degli elementi dell'insieme da rappresentare e poi tradurre in binario tali valori.

Enumerazione per le carte da briscola

Per esempio, poiché il mazzo di carte da briscola è composto 10 carte per ognuno dei 4 semi, si può scegliere:

seme	carte	ordine	codifica
bastoni	asso, 2-7, fante, cavallo, re	0-9	000000-001001
coppe	asso, 2-7, fante, cavallo, re	10-19	001010-010011
denari	asso, 2-7, fante, cavallo, re	20-29	010100-011101
spade	asso, 2-7, fante, cavallo, re	30-39	011110-100111

L'enumerazione proposta sarebbe una buona codifica in decimale: la prima cifra rappresenta il seme (0-3) e la seconda il valore (0-9).

Codifica per le carte da briscola

Una codifica migliore può essere ottenuta utilizzando le prime due cifre binarie per rappresentare il seme, e le rimanenti per il valore:

seme	carte	codifica
bastoni	asso, 2-7, fante, cavallo, re	00 0000 – 00 1001
coppe	asso, 2-7, fante, cavallo, re	01 0000 – 01 1001
denari	asso, 2-7, fante, cavallo, re	10 0000 – 10 1001
spade	asso, 2-7, fante, cavallo, re	11 0000 – 11 1001

Il vantaggio di questa codifica rispetto alla precedente è la facilità di codifica/decodifica.

Altra codifica per le carte da briscola

Un'altra codifica può essere ottenuta utilizzando le quattro cifre binarie per il valore in modo da codificare esplicitamente le carte che danno punti:

carte	codifica	carte	codifica
2	0000	fante	1000
4	0001	cavallo	1001
5	0010	re	1010
6	0011	tre	1011
7	0100	asso	1100

Riassumendo ...

Una carta è caratterizzata da seme e valore.

Ogni seme può assumere 4 configurazioni, mentre ogni valore può assumere 10 configurazioni.

oggetto	#config.	#bit
seme	4	2
valore	10	4
carta	$4 \times 10 = 40$	6

La codifica "seme+valore" usa lo stesso numero di bit della codifica "carta", ma non è una regola generale.

Codifica per le automobili

Un'auto è prodotta in 3 motorizzazioni e 10 colori.

Quanti bit sono necessari per descrivere un modello di questa automobile?

oggetto	#config.	#bit
motore	3	2
colore	10	4
automobile	$3 \times 10 = 30$	5

In questo caso, la codifica "motore+colore" è più ingombrante della codifica "automobile".

Regola generale

Sebbene:

$$\log_b mn = \log_b m + \log_b n$$

In generale:

$$\lceil \log_b mn \rceil \leq \lceil \log_b m \rceil + \lceil \log_b n \rceil$$

Una mano a briscola

D Quanti bit servono per codificare una mano di briscola?

R Una mano di briscola è composta da tre carte (diverse tra loro) prese da un mazzo di carte.

Una mano di briscola è quindi equivalente ad una combinazione di quaranta oggetti su tre posizioni.

Poiché $C(40, 3) = \frac{40!}{37! \cdot 3!} = \frac{40 \cdot 39 \cdot 38}{3 \cdot 2} = 9880$,
serviranno $\lceil \log_2 9880 \rceil = 14$ bit.

Nota: ripetere tre volte la codifica della singola carta richiede 18 bit.

Rappresentare elementi e insiemi (1)

D Gli oggetti sotto riportati sono elementi delle comuni interfacce grafiche. Come rappresentare in binario gli stati di tali oggetti?

<input type="radio"/>	Qui
<input checked="" type="radio"/>	Quo
<input type="radio"/>	Qua

Radio button

<input checked="" type="checkbox"/>	Qui
<input type="checkbox"/>	Quo
<input checked="" type="checkbox"/>	Qua

Check box

R I *radio button* permettono di selezionare uno solo degli n elementi della lista, mentre le *check box* consentono di selezionare un sottoinsieme degli n elementi della lista.

Rappresentare elementi e insiemi (2)

Quante configurazioni diverse può assumere un gruppo di n radio button?

Solo una voce può essere attiva.

Quindi, il numero di configurazioni sarà pari a n .

Pertanto, saranno necessari $\lceil \log_2 n \rceil$ bit.

Nel caso in esame, useremo 2 bit per i radio button.

Rappresentare elementi e insiemi (3)

Lo stato di un gruppo di check box equivale ad un sottoinsieme delle voci.

Il numero di sottoinsiemi di un insieme di n elementi è 2^n .

È necessario dedicare un bit per ogni voce per indicare se essa è attiva o no.

Pertanto, saranno necessari n bit.

Nel caso in esame, useremo 3 bit per i check box.

Rappresentare elementi e insiemi (4)

Enumeriamo le voci come di seguito riportato:

<input type="radio"/> Qui	0	<input checked="" type="checkbox"/> Qui
<input checked="" type="radio"/> Quo	1	<input type="checkbox"/> Quo
<input type="radio"/> Qua	2	<input checked="" type="checkbox"/> Qua

Possiamo individuare le seguenti codifiche:

Radio button: la posizione della voce attiva (1) si può codificare con la stringa binaria 01;

Check box: ponendo ad 1 i bit corrispondenti alle voci attive ed a 0 gli altri:

<i>posizione</i>	2	1	0
<i>valore</i>	1	0	1

Gelateria (1)

Una gelateria dispone di 12 gusti di gelato, 5 guarnizioni e 2 contenitori.

I gelati che vende sono sempre composti da un contenitore, 3 gusti e 2 guarnizioni.

I contenitori sono divisi in tre scomparti asimmetrici.

Le guarnizioni vengono distribuite uniformemente sul gelato.

Calcolare i bit necessari per codificare:

- a) i singoli elementi (gusti, guarnizioni, contenitori);
- b) un gelato.

Gelateria (2)

a) Codifica dei singoli elementi (gusti, guarnizioni, contenitori):

- 12 gusti $\rightarrow \lceil \log_2 12 \rceil = 4$ bit;
- 5 guarnizioni $\rightarrow \lceil \log_2 5 \rceil = 3$ bit;
- 2 contenitori $\rightarrow \lceil \log_2 2 \rceil = 1$ bit.

Gelateria (3)

- b) Codifica del gelato (3 gusti, 2 guarnizioni, 1 contenitore).

I gusti possono essere ripetuti e l'ordine è importante:

$$D_r(12, 3) = 12^3 \text{ configurazioni.}$$

Anche le guarnizioni possono essere ripetute, ma non importa l'ordine:

$$C_r(5, 2) = C(6, 2) = 15 \text{ configurazioni.}$$

Il contenitore può essere solo uno:
2 configurazioni.

Perciò si possono avere $12^3 \cdot 15 \cdot 2$ possibili gelati.

Gelateria (4)

- b) Codifica del gelato (3 gusti, 2 guarnizioni, 1 contenitore).

Con qualche passaggio matematico:

$$12^3 \cdot 15 \cdot 2 = 2^7 \cdot 3^3 \cdot 15 = 2^7 \cdot 405$$

Serviranno quindi:

$$\begin{aligned} \lceil \log_2 2^7 \cdot 405 \rceil &= \lceil \log_2 2^7 + \log_2 405 \rceil \\ &= \lceil 7 + \log_2 405 \rceil = 7 + \lceil \log_2 405 \rceil = 7 + 9 = 16 \text{ bit.} \end{aligned}$$