
Fondamenti di Informatica
per la Sicurezza
a.a. 2003/04

◇ ***Esercizi sulla codifica binaria*** ◇

Stefano Ferrari



Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione

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

Risposta Ipotizzando che ogni dito possa assumere solo due posizioni (disteso o chiuso), si ha a disposizione un numero di configurazioni pari a $2^{10} = 1024$.
Si può quindi contare da 1 a 1024 (o da 0 a 1023!).

Dimensione di una immagine digitale

Domanda Quanti bit servono per rappresentare un'immagine di uno schermo 1024×768 a colori, 8 bit per canale colore?

Risposta L'immagine è composta da $1024 \times 768 = 786432$ pixel. Ogni pixel è rappresentato da 3 colori (rosso, verde e blu) e ogni colore occupa 8 bit.

Una immagine così fatta, occupa quindi $786432 \times 3 \times 8 = 18.874.368$ bit.

Essi equivalgono a $18.874.368/8 = 2.359.296$ byte.

Esprimibili anche come $2.359.296/1024 = 2304$ kiB (*kibibyte*) o $2304/1024 = 2,25$ MiB (*mebibyte*).

Una digressione: prefissi binari

Nota: l'uso del termine *kilobyte* per indicare $2^{10} = 1024$ byte, sebbene diffuso, è inaccurato e può generare confusione.

L'*International Electrotechnical Commission* (IEC) ha proposto l'utilizzo di prefissi quali *kibi*, *mebi* o *gibi* per indicare le potenze di 2 (2^{10} , 2^{20} e 2^{30}) che meglio approssimano le potenze di 10 (10^3 , 10^6 , 10^9) che nel sistema internazionale vengono indicate dai prefissi kilo, mega e giga.

Per maggiori informazioni:

<http://physics.nist.gov/cuu/Units/binary.html>

Domanda Quale può essere una buona codifica per un mazzo di carte da briscola?

Risposta Alcune cose da tenere a mente:

- Una codifica è una scelta arbitraria.
- Una codifica può essere considerata buona se facilita
 - l'elaborazione nella quale la si vuole utilizzare
 - le operazioni di codifica o decodifica
- Codificare in binario un insieme di n simboli richiede almeno $\log_2 n$ bit.

Non essendo segnalata nessuna elaborazione particolare, ogni codifica potrebbe essere buona. Poiché il mazzo di carte da briscola è composto da 40 carte, si dovranno utilizzare 6 bit ($5 \leq \log_2 40 \leq 6$, 5 bit sarebbero troppo pochi).

Codifica per le carte da briscola (2)

Uno dei modi di ottenere una codifica è individuare una enumerazione degli elementi dell'insieme da rappresentare e poi tradurre in binario i valori risultanti. 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 (3)

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 decodifica.

Domanda Convertire il numero 1001 espresso in base 10, nella corrispondente rappresentazione in base 3.

Risposta La conversione si effettua iterando il processo di divisione intera per 3 e conservando il resto della divisione.

quoziente	resto
1001	
333	2
111	0
37	0
12	1
4	0
1	1

Risulta quindi:

$$1001_{10} = 1101002_3$$

Conversione da ottale in binario

Domanda Come si rappresenta il numero 57_8 in binario?

Risposta Si può effettuare la conversione con l'algoritmo generale basato su divisione intera e resto.

Tuttavia, poiché 8 è una potenza di 2, la conversione da ottale a binario (e viceversa) è facilitata.

Infatti:

$$\begin{aligned} 57_8 &= 5 \times 8 + 7 \times 1 = \\ &= 5 \times 2^3 + 7 \times 2^0 = \\ &= 101_2 \times 2^3 + 111_2 \times 2^0 = \\ &= 101111_2 \end{aligned}$$

L'ultimo passaggio è motivato dalla proprietà di traslazione della rappresentazione posizionale: moltiplicare per una potenza della base equivale a traslare di tante cifre quante l'esponente.

Conversione da binario in ottale

Domanda Come si rappresenta il numero 10101110_2 in ottale?

Risposta Applicando il ragionamento inverso quello del precedente esercizio, si procede dividendo la rappresentazione binaria in gruppi di tre simboli e poi si converte ciascun gruppo in ottale.

$$\begin{aligned}10101110_2 &= 10_2 \times 2^6 + 101_2 \times 2^3 + 110_2 \times 2^0 = \\ &= 2_8 \times 8^2 + 5_8 \times 8^1 + 6_8 \times 8^0 = \\ &= 256_8\end{aligned}$$

Nota: Analoghi ragionamenti valgono per la rappresentazione esadecimale.

Rappresentare elementi e insiemi

Domanda Gli oggetti sotto riportati sono elementi delle comuni interfacce grafiche: un gruppo di radio button e uno di check box. 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

Risposta Sebbene gli oggetti sopra riportati presentino la stessa lista di voci, il loro funzionamento è molto differente. I *radio button* permettono di selezionare uno solo degli elementi della lista, mentre le *check box* consentono di selezionare un sottoinsieme degli elementi della lista.

Rappresentare elementi e insiemi (2)

La codifica dello stato di un gruppo di radio button richiede di rappresentare un numero di elementi pari al numero delle voci della lista ($\log_2 N$ bit), mentre per codificare lo stato di un gruppo di check box bisogna rappresentare la presenza o l'assenza di ciascuna voce (N bit). In altri termini, per rappresentare uno stato dei radio button basta la posizione di quello attivo, mentre per rappresentare lo stato delle check box è necessario dedicare un bit per ogni voce, per indicare se è attiva o no. Enumerando 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

Rappresentare elementi e insiemi (3)

possiamo individuare le seguenti codifiche:

Radio button: usando 2 bit (2 è l'intero immediatamente superiore a $\log_2 3$) rappresentiamo la posizione della voce attiva (1), ottenendo:

01

Check box: usando 3 bit (il numero di voci della lista) poniamo ad 1 i bit corrispondenti alle voci attive ed a 0 quelli relativi alle voci disattive:

posizione 2 1 0

valore 1 0 1