Numerazione binaria
e rappresentazione delle informazioni

Info

Sito del corso:
http://home.dei.polimi.it/amigoni/InformaticaB.html

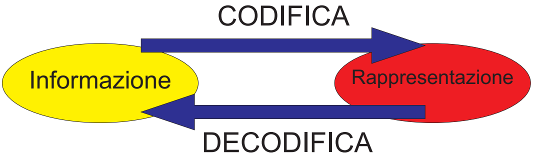
- Nicola Basilico, nicola.basilico@gmail.com

Problema

- Abbiamo informazioni (numeri, testi, immagini, suoni...) che vogliamo rappresentare (e poter elaborare) in un calcolatore.
- Vincolo: per motivi tecnologici un calcolatore lavora solo con i valori 0 e 1

Codifica e Decodifica

- Processo che permette di ottenere la rappresentazione delle informazioni



- Il processo inverso è la decodifica

Codifica e Decodifica

- L'elemento base della rappresentazione è il bit (binary digit, cifra binaria)
- Rappresentazione **binaria**
- Con un bit possiamo rappresentare un'informazione che può assumere 2 valori

Esempio

- Stato di una lampadina: $0 = \textit{spento}$, $1 = \textit{acceso}$
- Verità di una formula: $0 = \textit{falso}$, $1 = \textit{vero}$

Codifica e Decodifica

- Per rappresentare più valori è necessario usare **sequenze di bit**

1 bit	2 valori: 0, 1
2 bit	4 valori: 00, 01, 10, 11
3 bit	8 valori: 000, 001, 010, 011, 100, 101, 110, 111

- In generale: n bit rappresentano 2^n diversi valori
- 4 bit: nibble, 8 bit: byte

Codifica e Decodifica

Esempio di codifica

- Vogliamo dare una rappresentazione binaria per i quattro punti cardinali
- Ciascun punto cardinale può essere rappresentato da una **sequenza di 2 bit** secondo la seguente codifica

Nord	00
Est	01
Sud	10
Ovest	11

- Se avessimo voluto rappresentare i 4 semi di un mazzo di carte?

Codifica e Decodifica

- La codifica è una convenzione!
- E´ il modo in cui associamo un'informazione ad una sua rappresentazione binaria.

Codifica e Decodifica

Esercizio 1

- Domanda:
  quanti diversi valori posso rappresentare con 5 bit?
- Risposta:
  con 5 bit posso rappresentare $2^5 = 32$ diversi valori.

Esercizio 2

- Domanda:
  quanti diversi valori posso rappresentare con 2 byte?
- Risposta:
  2 byte $=$ 16 bit, quindi posso rappresentare $2^{16} = 65536$ diversi valori.

Codifica e Decodifica

Esercizio 3

- Domanda: quanti bit mi servono per rappresentare 1000 diversi valori?
- Risposta: devo trovare il minimo numero n di bit che soddisfi $2^n \geqslant 1000$, $2^{10} = 1024$, quindi $n = 10$.

Esercizio 4

- Domanda: quanti bit mi servono per rappresentare 112 diversi valori?
- Risposta: 7 bit ($2^7 = 128$). 6 bit sarebbero stati pochi, mentre 8 bit sarebbero stati troppi!

Codifica e Decodifica

- Attraverso meccanismi di codifica possiamo rappresentare diversi tipi di informazione:
  - Numeri naturali (insieme $\mathbb{N}$)
  - Numeri interi (insieme $\mathbb{Z}$)
  - Numeri razionali (insieme $\mathbb{Q}$)
  - caratteri
  - immagini
  - suoni
  - video
- Esistono diverse convenzioni (codifiche) per fornire a ciascun tipo di informazione una rappresentazione binaria.

Rappresentazione dei numeri naturali

Come rappresentare un numero naturale?

- Rappresentazione decimale posizionale
  - si utilizzano 10 cifre decimali (0, 1, ..., 9): la base è 10
  - *posizionale*: il significato di ogni cifra dipende dalla sua posizione relativa
  - le posizioni si contano da destra a sinistra partendo da 0

$$1919 = 1 \times 10^3 + 9 \times 10^2 + 1 \times 10^1 + 9 \times 10^0$$

Rappresentazione dei numeri naturali

- In generale: numero di n cifre in base b:

$$(a_{n-1}, \ldots, a_1, a_0) = a_{n-1} \times b^{n-1} + \ldots + a_1 \times b^1 + a_0 \times b^0 = \sum_{i=0}^{n-1} a_i \times b^i$$

- cifre usate: 0,1, . . . , b-1
- a_0 è la cifra **meno** significativa (LSD)
- a_{n-1} è la cifra **più** significativa (MSD)

Rappresentazione dei numeri naturali

- Rappresentazione **binaria**
  - base 2 ($b = 2$)
  - le cifre (binarie) sono: 0 e 1
- Rappresentazione **ottale**
  - base 8 ($b = 8$)
  - le cifre (ottali) sono: 0, 1,. . . , 7
- Rappresentazione **esadecimale**
  - base 16 ($b = 16$)
  - le cifre (esadecimali) sono: 0,1,. . . , 9, A, B, C, D, E, F

Rappresentazione dei numeri naturali - conversioni

Da base 2 a base 10:

$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = (5)_{10}$

$(10100)_2 = 1 \times 2^4 + 1 \times 2^2 = 16 + 4 = (20)_{10}$

Da base 8 a base 10:

$(12)_8 = 1 \times 8^1 + 2 \times 8^0 = 8 + 2 = (10)_{10}$

$(205)_8 = 2 \times 8^2 + 5 \times 8^0 = 128 + 5 = (133)_{10}$

Da base 16 a base 10:

$(2F)_{16} = 2 \times 16^1 + 15 \times 16^0 = 32 + 15 = (47)_{10}$

$(31A)_{16} = 3 \times 16^2 + 1 \times 16^1 + 10 \times 16^0 = 768 + 16 + 10 = (794)_{10}$

Rappresentazione dei numeri naturali - conversioni

Esercizio:
convertire in base 10 i seguenti numeri

$$
\begin{array}{ccc}
(1101010)_2 & (1110001)_2 & (1010101110)_2 \\
(145)_8 & (7342)_8 & (12345)_8 \\
(AF4)_{16} & (FF5E)_{16} & (ADC2D)_{16}
\end{array}
$$

Rappresentazione dei numeri naturali - conversioni

- Come convertire un numero da base 10 a base 2, 8 o 16?

Procedimento

Abbiamo un numero $(n)_{10}$ da convertire nella base b:

1. dividere n per b con una divisione intera
2. il resto della divisione diventa la cifra meno significativa (la prima che resta da calcolare) del numero in base b
3. se il quoziente è 0 abbiamo finito
4. se il quoziente è diverso da zero si torna al passo 1 considerando il quoziente come dividendo

Rappresentazione dei numeri naturali - conversioni

Esempio: $(13)_{10} = (1101)_2$

$$\begin{array}{lll} 13 : 2 = 6 & \text{resto} = 1 & \text{LSD} \\ 6 : 2 = 3 & \text{resto} = 0 & \\ 3 : 2 = 1 & \text{resto} = 1 & \\ 1 : 2 = 0 & \text{resto} = 1 & \text{MSD} \end{array}$$

Esempio: $(63)_{10} = (111111)_2$

$$\begin{array}{lll} 63 : 2 = 31 & \text{resto} = 1 & \text{LSD} \\ 31 : 2 = 15 & \text{resto} = 1 & \\ 15 : 2 = 7 & \text{resto} = 1 & \\ 7 : 2 = 3 & \text{resto} = 1 & \\ 3 : 2 = 1 & \text{resto} = 1 & \\ 1 : 2 = 0 & \text{resto} = 1 & \text{MSD} \end{array}$$

Rappresentazione dei numeri naturali - conversioni

Esempio: $(49)_{10} = (61)_8$

$$\begin{array}{lll} 49 : 8 = 6 & \text{resto} = 1 & \text{LSD} \\ 6 : 8 = 0 & \text{resto} = 6 & \text{MSD} \end{array}$$

Esempio: $(251)_{10} = (FB)_{16}$

$$\begin{array}{lll} 251 : 16 = 15 & \text{resto} = 11 = \text{B} & \text{LSD} \\ 15 : 16 = 0 & \text{resto} = 15 = \text{F} & \text{MSD} \end{array}$$

Rappresentazione dei numeri naturali - conversioni

Esercizio: convertire dalla base 10 i seguenti numeri

$$
\begin{array}{ccc}
(59)_{10} \rightarrow (?)_2 & (149)_{10} \rightarrow (?)_2 & (1387)_{10} \rightarrow (?)_2 \\
(77)_{10} \rightarrow (?)_8 & (132)_{10} \rightarrow (?)_8 & (1211)_{10} \rightarrow (?)_8 \\
(34)_{10} \rightarrow (?)_{16} & (112)_{10} \rightarrow (?)_{16} & (3459)_{10} \rightarrow (?)_{16}
\end{array}
$$

Rappresentazione dei numeri naturali - conversioni

- Come convertire un numero da base 2 a base 8?
- Raggruppiamo il numero binario a gruppi di **tre** cifre
- Convertiamo in sequenza ciascun gruppo in ottale

$$(000\ 100\ 111\ 001)_2 = (0471)_8$$

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Rappresentazione dei numeri naturali - conversioni

- Come convertire un numero da base 2 a base 16?
- Raggruppiamo il numero binario a gruppi di **quattro** cifre
- Convertiamo in sequenza ciascun gruppo in esadecimale

$$(0001\ 0011\ 1001)_2 = (139)_{16}$$

0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

Rappresentazione dei numeri naturali - somma

Somma tra numeri naturali binari:

$$\begin{array}{ll} 0+0=0 & \text{riporto} = 0 \\ 0+1=1 & \text{riporto} = 0 \\ 1+0=1 & \text{riporto} = 0 \\ 1+1=0 & \text{riporto} = 1 \end{array}$$

Esempio: $1101+111$

$$\begin{array}{cccccccc} \text{riporto} & 1 & 1 & 1 & 1 & & \\ & & 1 & 1 & 0 & 1 & + \\ & & 0 & 1 & 1 & 1 & = \\ \hline \text{somma} & 1 & 0 & 1 & 0 & 0 & \end{array}$$

Rappresentazione dei numeri naturali - somma

Esercizio:
eseguire in binario le seguenti somme

$$
\begin{array}{ccc}
(124 + 98) & (44 + 87) & (132 + 71) \\
(145 + 43) & (22 + 22) & (123 + 45)
\end{array}
$$

Rappresentazione dei numeri naturali

- Con n bit si possono rappresentare 2^n diversi numeri naturali. Quali sono?
- **Dalla codifica utilizzata** si deduce che i numeri rappresentabili appartengono all'intervallo: $(0; 2^n - 1)$

Rappresentazione dei numeri interi modulo e segno

Come rappresentare un numero intero con segno?

- Rappresentazione in **modulo e segno**
  - MSD indica il segno: $1 =$ negativo, $0 =$ positivo
  - i restanti bit indicano il modulo
  - esempio: $(1101)_2 = (-5)_{10}$, $(0111)_2 = (+7)_{10}$
  - notazione intuitiva ma...
  - scomoda per operazioni aritmetiche, lo zero ha due rappresentazioni!
- In questa codifica, con n bit si rappresentano i valori nell'intervallo: $(-2^{n-1} + 1; 2^{n-1} - 1)$

Rappresentazione dei numeri interi complemento a 2

Serve una rappresentazione che faciliti lo svolgimento delle operazioni:

- Rappresentazione in **complemento a 2**
  - Dati n bit, un numero positivo N è rappresentato in modo standard (come abbiamo visto per i naturali)
  - $-N$, invece si rappresenta come $2^n - N$
- Metodo operativo per rappresentare $-N$:
  - rappresentare il modulo N in modo standard
  - complementare tutti i bit ($1 \rightarrow 0$, $0 \rightarrow 1$)
  - sommare 1

Rappresentazione dei numeri interi complemento a 2

- In complemento a 2, con n bit, possiamo rappresentare gli interi nell'intervallo: $(-2^{n-1}; 2^{n-1} - 1)$
- Esempio: con 3 bit rappresentiamo i numeri in $(-4; 3)$

decimale	binario in C2
-4	100
-3	101
-2	110
-1	111

decimale	binario in C2
0	000
$+1$	001
$+2$	010
$+3$	011

- Il primo bit indica ancora il segno
- Lo zero ha una sola codifica

Rappresentazione dei numeri interi complemento a 2

Esercizio: convertire in complemento a 2 i seguenti numeri

$$\begin{array}{ccc} (12)_{10} & (-12)_{10} & (-8)_{10} \\ (1)_{10} & (-101)_{10} & (-54)_{10} \end{array}$$

Rappresentazione dei numeri interi complemento a 2

Come passare da complemento a 2 a base 10?

- Algoritmo inverso:
  - sottrarre 1
  - complementare a 1
  - convertire da binario a decimale e aggiungere il segno
- Metodo facilitato di verifica
  - convertire in decimale con l'algoritmo standard assegnando al bit più significativo valore negativo
  - Esempio: $(10100)_2 = -1 \times 2^4 + 1 \times 2^2 = (-12)_{10}$

Rappresentazione dei numeri interi
complemento a 2

Esercizio: riconvertire in decimale i seguenti numeri in complemento a 2

$$(1001010) \quad (011) \quad (1101001)$$
$$(11111) \quad (10100) \quad (101)$$

Rappresentazione dei numeri interi - somma

Somma tra numeri naturali interi con segno:

- rappresentare i numeri in complemento a 2
- effettuare la somma in modo standard
- non considerare l'eventuale riporto sul bit di segno

Esempio: $60 - 54 = 60 + (-54)$

riporto	1	1	1	1					
		0	1	1	1	1	0	0	$+$
		1	0	0	1	0	1	0	$=$
somma	(1)	0	0	0	0	1	1	0	

Rappresentazione dei numeri interi
Overflow

- Sommiamo due numeri in complemento a 2 rappresentati con n bit, quindi appartenenti a $(-2^{n-1}; 2^{n-1} - 1)$
- Può succedere che il risultato cada al di fuori dell'intervallo
- In altre parole: n bit non bastano per rappresentare il risultato! $\rightarrow$ **OVERFLOW**
- Come riconoscerlo?
  - può succedere solo quando si sommano due operandi dello stesso segno: **se il segno del risultato è diverso da quello degli operandi** è avvenuto un overflow!
  - gli ultimi due riporti sono diversi tra loro (01 o 10).

Esempio: $(100)_2 + (101)_2$

$$
\begin{array}{lcccccc}
\text{riporto} & 1 & 0 & & & \\
 & & 1 & 0 & 0 & + \\
 & & 1 & 0 & 1 & = \\
\hline
\text{somma} & (1) & 0 & 0 & 1 &
\end{array}
$$

Rappresentazione dei numeri frazionari

- Numeri frazionari: compresi tra 0 e 1

$$(0.a_{-1}a_{-2}\ldots a_{-n})_b = a_{-1}\times b^{-1} + a_{-2}\times b^{-2} + \ldots + a_{-n}\times b^{-n}$$

Esempio:

$$(0.587)_{10} = 5\times 10^{-1} + 8\times 10^{-2} + 7\times 10^{-3}$$

- Conversione da base 2 a base 10:

$$(0.1011)_2 = 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} + 1\times 2^{-4} = (0.6875)_{10}$$

Rappresentazione dei numeri frazionari

- Come convertire n da base 10 a base 2?

Procedimento

1. moltiplicare n per 2
2. la parte intera del risultato diventa la cifra **più** significativa (la prima che resta da calcolare) del numero in base 2
3. si torna al passo 1 considerando la parte frazionaria del risultato al posto di n

- Quando si finisce?
  - Soltanto i numeri del tipo $\frac{m}{2^z}$ possono essere rappresentati con un numero finito di cifre
  - Ci si ferma quando il numero di cifre calcolate costituisce un'approssimazione sufficiente

Rappresentazione dei numeri frazionari

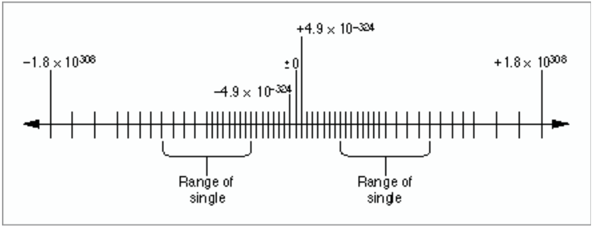
Esempio: $(0.587)_{10} = (?)_2$

$$
\begin{array}{lll}
0.587 \times 2 = 1.174 & \text{parte intera} = 1 & \text{MSD} \\
0.174 \times 2 = 0.348 & \text{parte intera} = 0 & \\
0.348 \times 2 = 0.696 & \text{parte intera} = 0 & \\
0.696 \times 2 = 1.392 & \text{parte intera} = 1 & \\
0.392 \times 2 = 0.784 & \text{parte intera} = 0 & \\
0.784 \times 2 = 1.568 & \text{parte intera} = 1 & \\
\quad \ldots & &
\end{array}
$$

Rappresentazione dei numeri razionali

- Rappresentazione in **virgola mobile**
- Un numero razionale N è espresso come: $N = m \times 10^e$
  - m è la **mantissa**, numero frazionario con segno
  - e è l'**esponente**, numero intero

Numeri rappresentabili in 64 bit



Rappresentazione dei caratteri

Tre possibili rappresentazioni:

- **ASCII standard**: un carattere è rappresentato con 7 bit *(ASCII = American Standard Code for Information Interchange)*
- **ASCII estesa**: un carattere è rappresentato con 8 bit
- **UNICODE** : un carattere è rappresentato con 16 bit (MS Windows ne usa una simile)

Rappresentazione dei caratteri

Pssiamo rappresentare caratteri, cifre, simboli, ...

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00100000	32	Spc	01000000	64	@	01100000	96	`
00100001	33	!	01000001	65	A	01100001	97	a
00100010	34	”	01000010	66	B	01100010	98	b
00100011	35	#	01000011	67	C	01100011	99	c
00100100	36	$	01000100	68	D	01100100	100	d
00100101	37	%	01000101	69	E	01100101	101	e
00100110	38	&	01000110	70	F	01100110	102	f
00100111	39	’	01000111	71	G	01100111	103	g
00101000	40	(	01001000	72	H	01101000	104	h
00101001	41	)	01001001	73	I	01101001	105	i
00101010	42	*	01001010	74	J	01101010	106	j
00101011	43	+	01001011	75	K	01101011	107	k
00101100	44	,	01001100	76	L	01101100	108	l
00101101	45	-	01001101	77	M	01101101	109	m
00101110	46	.	01001110	78	N	01101110	110	n
00101111	47	/	01001111	79	O	01101111	111	o
00110000	48	0	01010000	80	P	01110000	112	p
00110001	49	1	01010001	81	Q	01110001	113	q
00110010	50	2	01010010	82	R	01110010	114	r
00110011	51	3	01010011	83	S	01110011	115	s
00110100	52	4	01010100	84	T	01110100	116	t
00110101	53	[illegible]	01010101	85	[illegible]	01110101	117	[illegible]

Rappresentazione dei caratteri

E le parole? Sono sequenze di caratteri

Esempio: **informatica generale**

01101001	01101110	01100110	01101111	01110010
i	n	f	o	r
01101101	01100001	01110100	01101001	01100011
m	a	t	i	c
01100001	00100000	01100111	01100101	01101110
a		g	e	n
01100101	01110010	01100001	01101100	01100101
e	r	a	l	e

Rappresentazione dei caratteri

Esercizio 1

- Domanda:
  alfabeto immaginario di 322 simboli: quanti bit per rappresentarli tutti?
- Risposta:
  $2^9 \geqslant 322$, $2^9 = 512$, quindi $n = 9$

Esercizio 2

- Domanda:
  quanti bit o byte occupa la frase *biologia marina* nelle tre diverse codifiche?
- Risposta:
  In ASCII 105bit, in ASCII estesa 15byte, in UNICODE 30byte.

Codifica

Esercizio

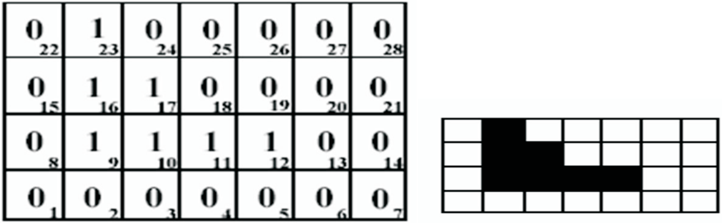
- Domanda:
  Cosa rappresenta la stringa 01000011?
- Risposta:
  Se interpretata come un numero naturale è 67, se interpretata come carattere è C. Dipende!

- La codifica è una convenzione!

Rappresentazione delle immagini

L'immagine è suddivisa in pixel

Ad ogni pixel associamo una rappresentazione binaria:



0000000111100011000000100000

Rappresentazione delle immagini

- Assegnando un bit ad ogni pixel si possono rappresentare solo immagini in bianco e nero
- Per rappresentare immagini a diversi livelli di grigio o a colori: a ogni pixel è associata una sequenza di bit
  - con 8 bit per pixel: $2^8 = 256$ livelli di grigio
  - con 24 bit per pixel: $2^{24} = 16777216$, 16.7 milioni di colori

Rappresentazione delle immagini

- Nei monitor è utilizzato lo **standard RGB**: ogni colore è ottenuto mescolando tre diverse gradazioni dei colori primari (rosso verde e blu)
- Per ogni pixel bisogna specificare quali sono i livelli dei tre colori
- Esempio: un byte per ogni livello. Un pixel è rappresentato con 24 bit (3 byte).
- Risoluzione: numero di pixel presenti sullo schermo (800×600, 1024×768, 1600×1200)

Rappresentazione delle immagini

Esercizio 1

- Domanda:
  quanti byte occupa un'immagine di 100×100 pixel in bianco e nero?
- Risposta:
  l'immagine è composta da $100 \times 100 = 10000$ pixel. Per ogni pixel, in bianco e nero, serve 1 bit quindi servono in totale 10000 bit e cioè $10000/8 = 1250$ byte.

Esercizio 2

- Domanda:
  quanti byte occupa un'immagine di $100x100 pixel$ a 256 colori?
- Risposta:
  l'immagine è composta da 10000 pixel. Per ogni pixel, con 256 colori, serve 1 byte (8 bit), quindi servono in totale 10000 byte.

Esercizio 3

- Domanda:
  se un'immagine a 16,7 milioni di colori occupa 2400 byte, da quanti pixel sarà composta?
- Risposta:
  con 16,7 milioni di colori un pixel occupa 3 byte, quindi l'immagine occupa $2400/3 = 800$ pixel

Rappresentazione delle immagini

- Immagini **bitmap**
  - rappresentate pixel per pixel
  - tipicamente in file con estensione .bmp
  - hanno elevate dimensioni
- Immagini **bitmap compresse**
  - GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group)
  - Per esempio, se k pixel lungo la stessa riga hanno lo stesso colore, si memorizza il colore una volta sola e il numero k
- Immagini **vettoriali**
  - sono rappresentate specificando gli elementi geometrici (punti, segmenti, poligoni,...) che le compongono
  - SVG (Scalable Vector Graphics)
  - dimensioni ridotte

Unità di misura

Di solito si usano i multipli del byte

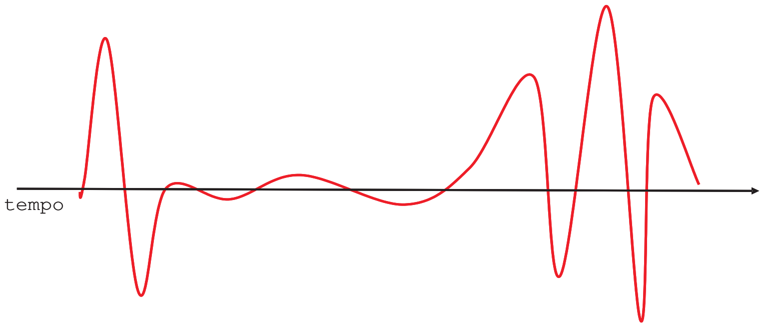
Kilo	KB	2^{10} (circa un migliaio, 1024)
Mega	MB	2^{20} (circa un milione, $1KB \times 1024$)
Giga	GB	2^{30} (circa un miliardo, $1MB \times 1024$)
Tera	GB	2^{40} (circa un migliaio di miliardi, $1GB \times 1024$)

Rappresentazione di video

- Un filmato è una sequenza temporale di immagini, dette **frames**
- Per rappresentare un filmato si digitalizzano i suoi frames
- Vari formati
  - .avi (Audio Video Interleave, Microsoft)
  - .mov (anche detto QuickTime, Apple)
  - .mpeg (anche detto QuickTime, Apple)
  - DivX ;-)
- Compressione: rappresentare solo differenze tra frame successivi

Rappresentazione dei suoni

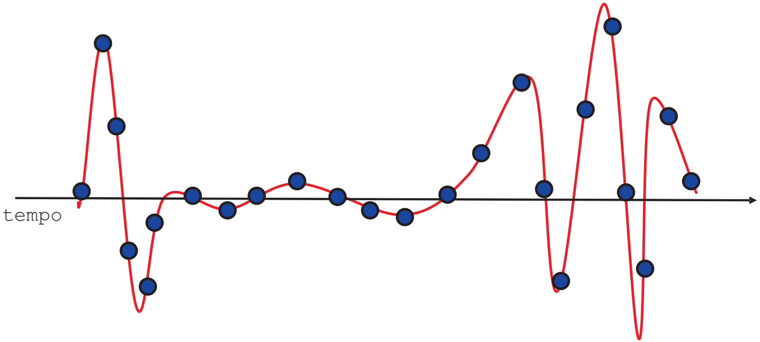
Fisicamente un suono è rappresentato come un'onda che descrive la variazione della pressione dell'aria nel tempo (onda sonora):



Rappresentazione dei suoni

L'onda sonora è campionata, cioè si misura l'ampiezza ad intervalli di tempo regolari:

- Ampiezza misurata in un dato istante di tempo = campione
- Il numero di misure effettuate in un secondo definisce la frequenza di campionamento (Hertz, Hz)



Rappresentazione dei suoni

- L'accuratezza della ricostruzione dipende
  - dalla frequenza di campionamento
  - dal numero di bit usati per rappresentare ogni campione
- Maggiore accuratezza significa maggior quantità di memoria occupata
- Tecniche di compressione
  - Algoritmi lossy: sfruttano il fatto che suoni a basso volume sovrapposti a suoni ad alto volume sono poco udibili dall'orecchio umano e possono essere eliminati
  - MPEG-1 Layer 3, detto anche MP3

Rappresentazione dei suoni

Esercizio 1

- Domanda:
  quanto spazio occupa un suono della durata di 10 secondi campionato a 100 Hz, in cui ogni campione occupa 4 byte?
- Risposta:
  la frequenza di campionamento ci dice quanti campioni di suono vengono memorizzati in un secondo, 100 in questo caso. Avendo 10 secondi di suono avremo $10 \times 100 = 1000$ campioni. Poichè ogni campione richiede 4 byte, il suono occuperà $1000 \times 4 = 4000$ byte

Rappresentazione dei suoni

Esercizio 2

- Domanda:
  un secondo di suono campionato a 512 Hz occupa 1KB.
  Quanti valori distinti si possono avere per i campioni?
- Risposta:
  poichè vengono memorizzati 512 campioni al secondo, avremo in tutto 512 campioni. Il file sonoro occupa 1 KB, cioè 1024 byte e quindi ogni singolo campione occuperà $1024/512 = 2$ byte, ovvero 16 bit. Si potranno quindi avere $2^{16} = 65536$ valori distinti.

Rappresentazione di informazioni complesse

- Per rappresentare un testo, un'immagine, un suono, la filosofia di base è la stessa
- Si stabilisce e si usa una prima convenzione per rappresentare ogni unità elementare (un carattere per il testo, un pixel per l'immagine, un campione per il suono)
- Si stabilisce e si usa una seconda convenzione per rappresentare insiemi di unità elementari
    - Tecniche di compressione