

Rappresentazione delle informazioni

Elementi di Informatica 2018/2019

Informazione

Informazione deriva da `informare` (dare forma)

Messaggio da `sorgente` a `destinatario`

`Protocollo` che definisce un insieme di regole da utilizzare

Rappresentazione: utilizzo di un `codice`

Analogico e Digitale

Rappresentazione analogica

Varia in analogia con una grandezza reale (tipo continuo)

Rappresentazione digitale

Insieme finito di elementi distinti (tipo discreto)

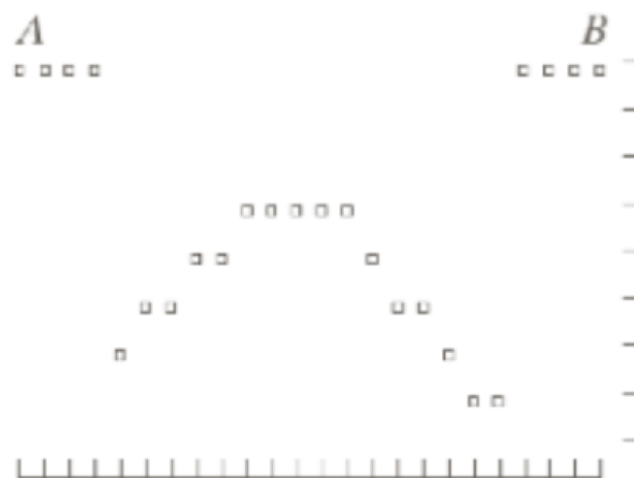
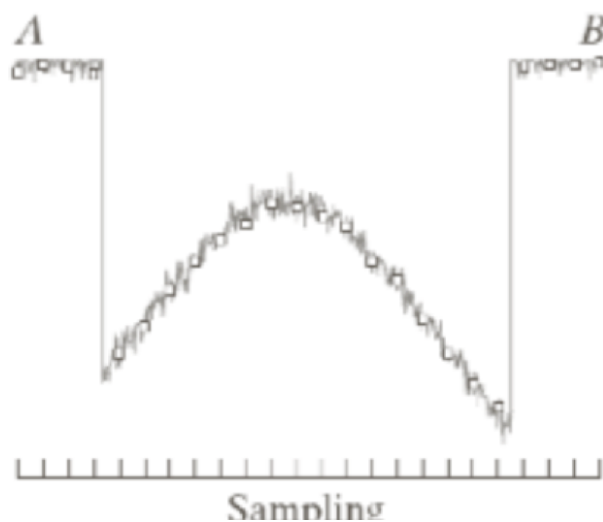
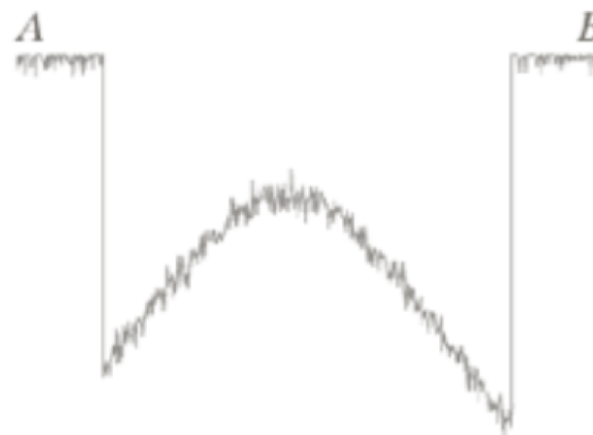
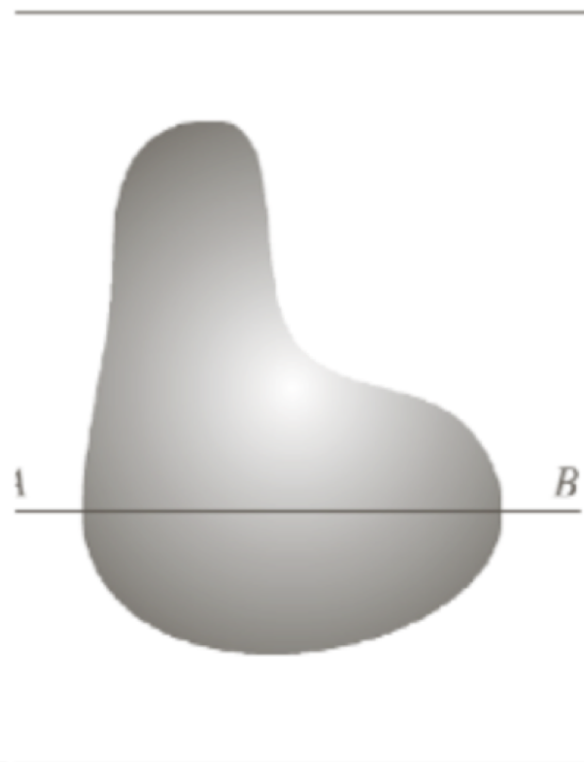
Da analogico a digitale

- Campionamento: da segnale continuo a segnale discreto
- Quantizzazione: da valori continui a valori discreti

a b
c d

FIGURE 2.16

Generating a digital image.
(a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



Codifica

Rappresentazione comprensibile dal calcolatore

INFORMAZIONE \leftrightarrow CODIFICA \leftrightarrow RAPPRESENTAZIONE

Codici

Insieme di simboli che permette di rappresentare dell'informazione

Alfabeto codice: 0, 1, ... , 9

Cardinalità (n): 10

Lunghezza parola codice (l)

N. di parole codice è dato da:

$$n^l$$

Esempio (base 10)

$$l = 1; n = 10; n^l; 10; 0 - 9$$

$$l = 2; n = 10; n^l; 100; 0 - 99$$

$$l = 3; n = 10; n^l; 1000; 0 - 999$$

Esempio (base 2)

$$l = 1; n = 2; n^l; 2; 0 - 1$$

$$l = 2; n = 2; n^l; 4; 0 - 11(0, 1, 10, 11)$$

$$l = 3; n = 2; n^l; 8; 0 - 111(0, 1, 10, 11, 100, 101, 110, 111)$$

Simboli e parole codice

$$n^l > m$$

dove m è il numero di parole che si vogliono rappresentare

Esempio

se $n = 10$, quanto deve valere l per poter rappresentare il numero 100?

Soluzione

se $l = 2$ il numero più alto rappresentabile è 99

Quindi serve $l = 3$

Ma con $l = 3$ si possono rappresentare i numeri fino al 999

Lunghezza fissa - lunghezza variabile

- Lunghezza fissa: tutte le parole hanno la stessa lunghezza (come nel computer)
- Lunghezza variabile: non tutte le parole hanno la stessa lunghezza (vedi vocabolario...)

Rappresentazione binaria

Rappresentazione basata sull'uso di **DUE** soli simboli: 0 e 1

E' la più piccola quantità di informazione rappresentabile su un elaboratore: **bit**

Rappresentazione binaria

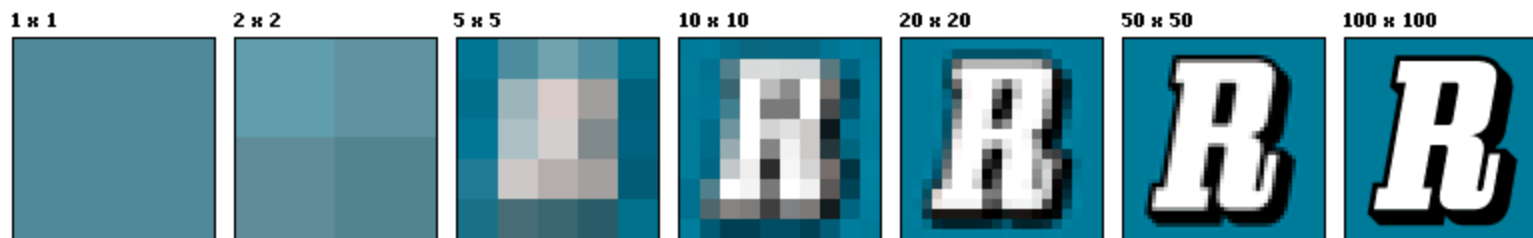
Perché binaria?

Rende un sistema digitale meno soggetto ad **errori** dovuti a disturbi elettrici

Flip-flop: supporto di memorizzazione che supporta due stati

Da analogico a digitale

Introduzione di un certo livello di approssimazione



Bit e byte

Per questioni "costruttive" generalmente si fa riferimento a blocchi di 8 bit detti **byte**

B	Byte	1	8
KB	KiloByte	$2^{10}=1024$	8.192
MB	MegaByte	$2^{20}=1.048.576$	8.388.608
GB	GigaByte	$2^{30}=1.073.741.824$	8.589.934.592
TB	TeraByte	$2^{40}=1.099.511.627.776$	8.796.093.022.208

Codifica nei calcolatori

Codifica binaria a lunghezza fissa: parole di lunghezza multipli di 8

Numero di byte b	Numero di bit ($b*8$)	$2^{(b*8)}$	Configurazioni
1	8	2^8	256
2	16	2^{16}	65.536
3	24	2^{24}	16.777.216
4	32	2^{32}	4.294.967.296

Rappresentazione dei numeri

Sistema di numerazione: insieme di simboli e regole che permettono di associare ad una sequenza *un numero*

- Posizionali
- Non posizionali

Sistema posizionale

E' caratterizzato da una **base** (n. di simboli a disposizione)

Utilizziamo un sistema posizionale in base 10

Decimale - Base 10: 0, ..., 9

Ottale - Base 8: 0, ..., 7

Esadecimale - Base 16: 0, ..., 9, A, B, C, D, E, F

Binario - Base 2: 0, 1

Sistema posizionale

$$N = c_{i-1} * b^{i-1} + ... + c_2 * b^2 + c_1 * b^1 + c_0 * b^0$$

Esempio:

$$21_{10} = 1 * 10^0 + 2 * 10^1 = 1 + 20 = 21$$

Sistema posizionale

Base 2

$$10101_2 = 1 * 2^0 + 1 * 2^2 + 1 * 2^4 = 1 + 4 + 16 = 21$$

Avendo la base 2 un numero di simboli minore rispetto alla base 10, la rappresentazione di uno stesso numero ha necessità di più cifre

Ottale ed Esadecimale

Per evitare di dover utilizzare stringhe di bit troppo lunghe si utilizzano i sistemi in **base 8** e **base 16**

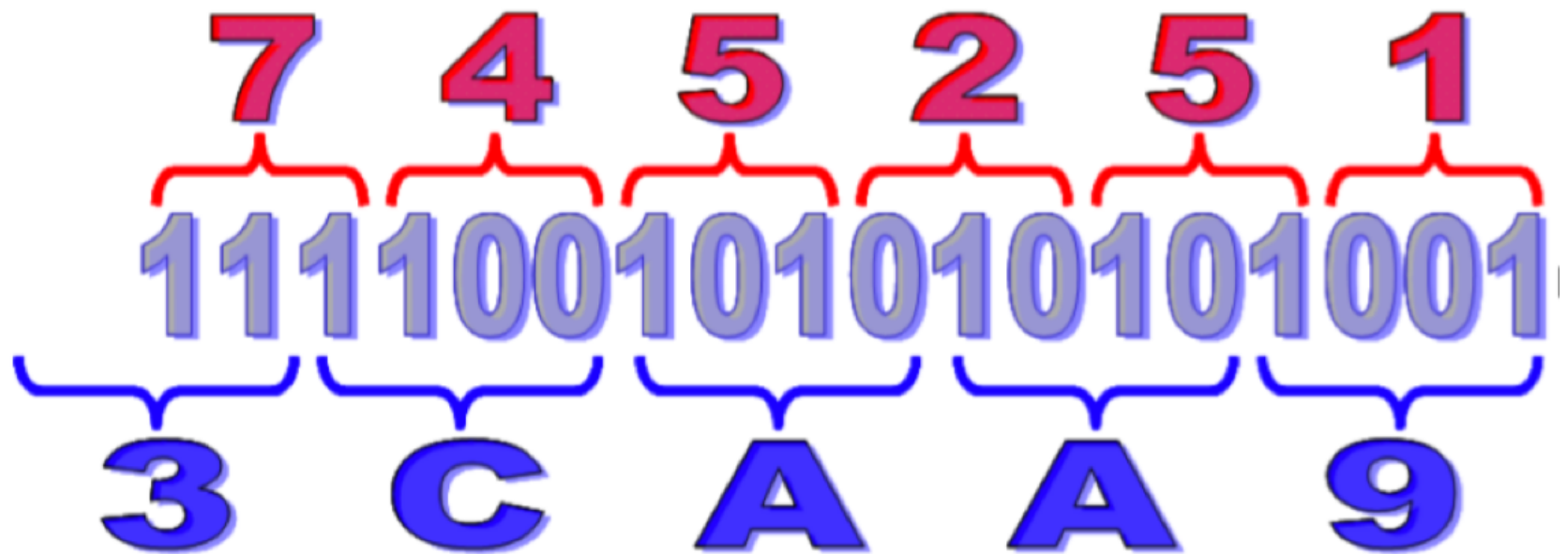
Ottale

Ottale	Binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Esadecimale

Esadecimale	Binario
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Conversione



Conversione

Decimale -> Binario

Dividere per due il numero da converire, annotare il resto e andare avanti fino a quando il quoziente è pari a zero.

Scrivere i resti in sequenza inversa.

Conversione: da decimale a binario

Esempio:

$$21_{10}$$

$$21/2 = 10 - \textit{resto} : 1$$

$$10/2 = 5 - \textit{resto} : 0$$

$$5/2 = 2 - \textit{resto} : 1$$

$$2/2 = 1 - \textit{resto} : 0$$

$$1/2 = 0 - \textit{resto} : 1$$

$$21_{10} = 10101_2$$

Codifica dei caratteri

Vengono rappresentati attraverso sequenze di *bit* utilizzando uno specifico codice: per esempio **ASCII**

ASCII a 7 bit: 128 caratteri

ASCII esteso a 8 bit: 256 caratteri

UNICODE a 16 o 32 bit: 65536 caratteri

Codifica delle immagini

Vengono rappresentate attraverso una sequenza di bit: *digitalizzazione*

- Pixel: definito da coordinate spaziali
- Valori: intensità del colore associato ad ogni pixel (potenza di 2)

Risoluzione: capacità di discriminare un dettaglio (nello spazio/nei valori di intensità)

Codifica delle immagini: standard di codifica

Tra i più diffusi: *TIFF*, *JPEG* e *PNG*

Compressione: *lossy* o *lossless*