



Metodologie informatiche per l'organizzazione dei servizi turistici

(Prof. Nicola Orio)

Rappresentazione dell'informazione

Corso di laurea in
Progettazione e gestione del turismo culturale



Codifica dell'informazione

L'informazione si presenta in forme diverse

- Il valore numerico di una grandezza fisica
- Il testo di un articolo di giornale
- Il suono prodotto da uno strumento musicale
- L'immagine di una fotografia
- Le sequenze video di una ripresa televisiva

La rappresentazione viene definita codifica

- ✓ Rappresentazione adeguata per ogni forma
- ✓ Tutte basate sulle sequenze finite di bit



Codifica di numeri

I numeri sono stati i primi ad essere codificati

- ✓ I computer facevano solo calcoli ripetitivi
 - Il termine “digitale” deriva da digit=cifra, numero

Com'è noto, i numeri si dividono in:

- ✓ Interi, razionali (aperiodici e periodici), irrazionali
 - Razionali aperiodici e irrazionali devono essere approssimati

In questo corso tratteremo solo i numeri naturali

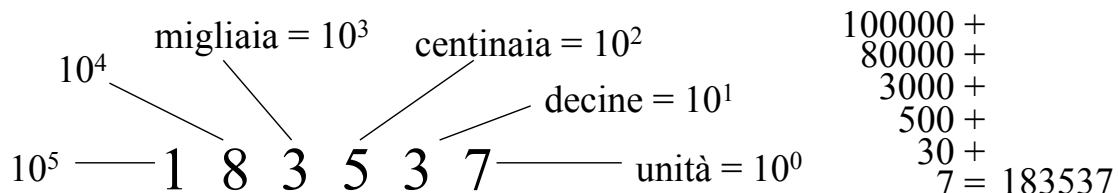
– Lo 0 e gli interi positivi



Rappresentazione in base 10

Caratteristiche del sistema decimale

- ✓ Basato sulla combinazione di 10 simboli (da 0 a 9)
- ✓ Il significato di ogni cifra dipende dalla sua posizione
 - Le cifre, partendo da destra, sono moltiplicate per una potenza crescente di 10 (1, 10, 100, ...)
 - I prodotti sono sommati tra loro





Rappresentazione in base 2

Simile quella decimale, con 2 al posto di 10

- ✓ Basato sulla combinazione di 2 simboli (0 e 1)
- ✓ Il significato di ogni cifra dipende dalla sua posizione
 - Le cifre, partendo da destra, sono moltiplicate per una potenza crescente di 2 (1, 2, 4, 8, ...)
 - I prodotti sono sommati tra loro

$$\begin{array}{ccccccc}
 & & 2^3 = 8 & & 2^2 = 4 & & 2^1 = 2 \\
 & 2^4 = 16 & & & & & \\
 & & \diagdown & / & \diagdown & / & \\
 2^5 = 32 & \text{---} & 1 & 0 & 1 & 1 & 0 & 1 & \text{---} & 2^0 = 1
 \end{array}$$

$$\begin{array}{r}
 1 * 32 + \\
 0 * 16 + \\
 1 * 8 + \\
 1 * 4 + \\
 0 * 2 + \\
 1 * 1 = 45
 \end{array}$$



Rappresentazione in base 16

Stesso principio di base 10 e base 2

- ✓ Si usano 16 simboli diversi
 - Cifre da 0 a 9
 - Lettere da A a F
- ✓ Usata spesso per rappresentare i colori
 - Un byte è codificato con 2 cifre esadecimali

$$\begin{array}{ccccccc}
 & & 16^3 = 4096 & & 16^2 = 256 & & 16^1 = 16 \\
 & 16^4 = 65536 & & & & & \\
 & & \diagdown & / & \diagdown & / & \\
 & & 3 & F & A & 0 & 8 & \text{---} & 16^0 = 1
 \end{array}$$

$$\begin{array}{r}
 3 * 65536 + \\
 F * 4096 + \\
 \{15 * 4096 +\} \\
 A * 256 + \\
 \{10 * 256 +\} \\
 0 * 16 + \\
 8 * 1 = 260616
 \end{array}$$



Conversioni tra le basi 2, 10, 16

<i>base 2</i>	<i>base 10</i>	<i>base 16</i>	<i>base 2</i>	<i>base 10</i>	<i>base 16</i>
0	0	0	1111	15	F
1	1	1	10000	16	10
10	2	2	10001	17	11
11	3	3	10010	18	12
100	4	4	11110	30	1E
101	5	5	11111	31	1F
110	6	6	100000	32	20
111	7	7	100001	33	21
1000	8	8	100010	34	22
1001	9	9	111111	63	3F
1010	10	A	1000000	64	40
1011	11	B	1111111	127	4F
1100	12	C	10000000	128	80
1101	13	D	11111111	255	FF
1110	14	E	100000000	256	100



Sequenze di numeri – 1

Normalmente si usa un simbolo separatore

– Le serie 14 7 1789 o 14-7-1789 o 14/7/1789 sono diverse dal numero 1471789

- ✓ Un bit può rappresentare solamente due simboli
 - Non si può rappresentare il simbolo separatore

Usare un numero costante di bit per ogni numero

- ✓ Indipendentemente da quanti sono necessari
 - Ad esempio la sequenza 100101110101 equivale a
 - 100, 101, 110, 101 = 4, 5, 6, 5 utilizzando tre bit per numero
 - 1001, 0111, 0101 = 9, 7, 5 utilizzando quattro bit per numero



Sequenze di numeri – 2

Utilizzare sequenze fisse di bit ha due svantaggi

- ✓ Non si possono rappresentare valori troppo elevati
 - Se si sfora il valore finale non è significativo
- ✓ Spesso si usano più bit del necessario
 - I bit più a sinistra sono uguali a zero
- ✓ E' necessario conoscere a priori il numero di bit

Normalmente si utilizzano byte e multipli

- 1 byte = 8 bit = numeri da 0 a 255
- 2 byte = 16 bit = numeri da 0 a 65535
- 4 byte = 32 bit = numeri da 0 a circa 4,3 miliardi
- 8 byte = 64 bit = numeri da 0 a circa 18 miliardi di miliardi



Codifica del testo

Un testo è una sequenza di simboli

- ✓ Organizzato in forma lineare
 - Concetto simile alla sequenza finita di bit

L'insieme di simboli è finito e noto a priori

- ✓ I simboli possono essere
 - Numeri, segni d'interpunzione, operatori
 - Lettere di un alfabeto
 - Latino, greco, cirillico, arabo, ebraico
 - Differenze nei diacritici (å ç ř ê)
 - Ideogrammi



Codifica di caratteri – 1

I caratteri appartengono ad un insieme noto

- ✓ Ad ogni carattere si associa un numero intero
 - L'intero è il codice del carattere
- ✓ L'associazione va fatta per tutti i possibili simboli
 - I simboli senza un codice non sono rappresentabili
 - Bisogna porre attenzione alla scelta dell'alfabeto
- ✓ L'associazione è arbitraria

Un testo diventa una sequenza di numeri

- ✓ Bisogna scegliere quanti bit si usano per ogni numero



Codifica di caratteri – 2

Il numero di bit dipende dall'alfabeto

- 6 bit = 64 caratteri
- 7 bit = 128 caratteri
- 8 bit = 256 caratteri
- 16 bit = 65536 caratteri

Non esiste una codifica naturale

- ✓ Conviene preservare l'ordine alfabetico
 - Ordinare un testo diventa come ordinare numeri
- ✓ E' necessario condividere l'associazione
 - Cambiando computer il testo può diventare illeggibile



ASCII a 7 bit – 1

Primo standard per il testo

- ✓ Utilizza 7 bit per carattere
 - Può rappresentare 128 caratteri diversi
- ✓ Codifica
 - Le lettere dell'alfabeto inglese
 - Non sono previsti diacritici
 - Tutti i tasti delle tastiere americane
 - Codici per il controllo delle stampanti
 - Ritorno di carrello, avanzamento di una linea, tabulazioni
- ✓ Vasta diffusione a livello internazionale



ASCII a 7 bit – 2

- ✓ I caratteri dell'ASCII a 7 bit sono gli unici ad essere universalmente riconosciuti da tutti i sistemi
 - I codici evidenziati sono codici di controllo
 - Molti sono obsoleti ma mantenuti per compatibilità

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



ASCII a 8 bit – 1

La codifica ASCII è stata estesa

- ✓ Rappresentazione a 8 bit (1 byte)
 - Codifica 256 caratteri (il doppio)
- ✓ Compatibilità con ASCII a 7 bit
 - I 256 codici divisi in due gruppi
 - I primi 128 codici immutati
 - Gli altri 128 codici sono dedicati a nuovi caratteri
- ✓ Mancanza di accordo sui nuovi codici
 - Esistono diverse versioni dell'ASCII a 8 bit
- ✓ Gli ideogrammi non sono rappresentati



ASCII a 8 bit – 2

Esistono molte versioni di ASCII a 8-bit

- ISO-8859-1 (Latin 1): per le lingue dell'Europa occidentale
 - ISO-8859-2 (Latin 2): per le lingue dell'Europa orientale
 - ISO-8859-3/4 (Latin 3/4): per esperanto e maltese / paesi ex URSS
 - ISO-8859-5 (Cyrillic): per lingue basate sull'alfabeto cirillico
 - ISO-8859-6 (Arabic): per l'arabo
 - ISO-8859-7 (Greek): per il greco moderno, senza diacritici
 - ISO-8859-8 (Hebrew): per l'ebraico
 - ISO-8859-9 (Turkish): per il turco, quasi uguale al Latin 1
 - ISO-9959-10/11...16: altre varianti del Latin
- ✓ Ci sono versioni proprietarie (IBM) ancora diverse



ASCII a 8 bit – 3

Possibili problemi di incompatibilità

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

ISO-8859-1 (Latin1)

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

ISO-8859-2 (Latin2)

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

ISO-8859-3 (Latin3)

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

ISO-8859-4 (Latin4)



Unicode – 1

Scopo: dare un codice unico per ogni simbolo

- ✓ Indipendentemente dal computer e dal software
 - Superare i limiti dell’ASCII
- ✓ Codifica tutte le lingue
 - Ideogrammi ma anche simboli di lingue morte
- ✓ Utilizza un sistema per minimizzarne la quantità
 - Un testo di solito è scritto in una sola lingua
 - Rappresenta solo un sottoinsieme dei caratteri di Unicode
 - Utilizza il numero di bit necessari per ogni lingua
- ✓ Esistono implementazioni a 8, 16 o 32 bit
 - Punto in comune con l’ASCII sono i primi 256 codici



Unicode – 2

C0 Controls and Basic Latin								Greek and Coptic								Hebrew							Katakana						
000	001	002	003	004	005	006	007	037	038	039	03A	03B	03C	03D	03E	03F	059	05A	05B	05C	05D	05E	05F	30A	30B	30C	30D	30E	30F
NUL	DLE	SP	0	@	P	`	p			ι	Π	ϖ	π	β	η	ξ								ニ	グ	ダ	バ	ム	キ
SOH	DC1	!	1	A	Q	a	q			Α	Ρ	α	ρ	Θ	λ	ε								ア	ケ	チ	パ	メ	エ
STX	DC2	"	2	B	R	b	r			Β		β	ς	Υ	Ω	Ϙ								ア	ゲ	ヂ	ヒ	モ	ヲ
ETX	DC3	#	3	C	S	c	s			Γ	Σ	γ	σ	Υ	Ω	ι								イ	コ	ツ	ビ	ヤ	ン
EOT	DC4	\$	4	D	T	d	t			Δ	Τ	δ	τ	Υ	Ω	θ								イ	ゴ	ツ	ピ	ヤ	ヴ
ENQ	NAC	%	5	E	U	e	u			Ε	Υ	ε	υ	Φ	Ϙ	€								ウ	サ	ツ	フ	ユ	カ
ACK	SYN	&	6	F	V	f	v			Α	Ζ	Φ	ζ	Ϙ	Ϙ	ε								ウ	ザ	テ	ブ	ユ	ケ
BEL	ETB	'	7	G	W	g	w			·	H	X	η	χ	ξ	δ								エ	シ	テ	プ	ヨ	ワ
BS	CAN	(8	H	X	h	x			Ε	Θ	Ψ	θ	υ	Q	Ϙ								エ	ジ	ト	ヘ	ヨ	ギ
HT	EM)	9	I	Y	i	y			H	I	Ω	ι	ω	Ϙ	C								オ	ス	ト	ベ	ラ	ゼ
LF	SUB	*	:	J	Z	j	z			I	K	İ	κ	ı	Σ	X								オ	ズ	ナ	ペ	リ	ヲ
VT	ESC	+	:	K	[k	{			Λ	Υ	λ	υ	ς	Σ	M								カ	セ	ニ	ホ	ル	・
FF	FS	.	<	L	\	l				O	M	ά	μ	ó	F	ρ								ガ	ゼ	ヌ	ボ	レ	ー
CR	GS	-	=	M]	m	}			N	É	v	ύ	F	ς	Ϙ								キ	ソ	ネ	ポ	ロ	、
SO	RS	>	N	^	n	~				Y	Ξ	ή	ξ	ω	Ζ	€								ギ	ゾ	ノ	マ	ワ	バ
SI	US	/	?	O	_	o	DEL			Ω	O	í	o		Ζ	†								ク	タ	ハ	ミ	ワ	Γ



Informazione continua – 1

Numeri e testo

- ✓ Sono rappresentati con simboli
 - Finiti noti a priori
 - Rappresentati da un'opportuna sequenza di bit

Suoni e immagini

- ✓ Sono di natura continua
- ✓ Infinite variazioni
 - Non è possibile una rappresentazione simbolica
 - Si deve approssimare la realtà



Informazione continua – 2

Nel mondo reale

- ✓ Le grandezze variano con continuità
 - Variazione nel tempo dei suoni
 - Variazione nello spazio delle immagini
- Necessario usare sequenze di lunghezza infinita
 - Si applica un campionamento
- ✓ Le grandezze assumono qualsiasi valore
 - Ampiezza dell'onda sonora o tonalità dei colori
- Necessario usare moltissimi bit
 - Si applica una quantizzazione



Campionamento – 1

Es. 1: La temperatura di una stanza

- ✓ Impossibile misurarla ad ogni istante
 - La temperatura varia lentamente
 - La misura a 12:00 è quasi uguale a 11:59 e 12:01
 - La misura a 12:00 simile a 11:45 e 12:15

Es. 2: L'altezza di un terreno

- ✓ Impossibile misurarla in ogni singolo punto
 - In molti casi i dislivelli sono minimi
 - La misura in un punto è quasi uguale nel centimetro intorno
 - La misura in un punto è simile a quella a 10 metri di distanza



Campionamento – 2

Data una grandezza che varia con continuità nel tempo o nello spazio

- ✓ Si sceglie ogni quanto misurarla
 - Si fa in modo che sia rappresentativa
 - Misure molto ravvicinate
 - Si devono fare tante misurazioni che occupano spazio
 - Misure molto distanziate
 - Si rischia di perdere variazioni importanti

Le misure si chiamano campioni



Campionamento – 3

In informatica i campioni sono equispaziati

- Non obbligatorio, scelto per semplicità
- ✓ Frequenza di campionamento
 - Numero di campioni nell'unità di tempo
 - Usato per i suoni
 - Numero di campioni nell'unità di lunghezza
 - Usato per le immagini
- ✓ L'informazione tra due campioni è persa del tutto
 - Campioni molto ravvicinati minimizzano la perdita
 - Compromesso tra precisione e costo (misurazione, spazio, ...)



Quantizzazione – 1

Es. 1: La temperatura di una stanza

- ✓ Il termometro è impreciso
 - Il valore oltre un certo numero di decimale non è significativo
- ✓ In casa non ha senso misurare i centesimi di grado
 - Sono inutili per la particolare applicazione

Es. 2: L'altezza di un terreno

- ✓ L'altimetro è impreciso
 - Non ha senso misurare i millimetri di altezza sul livello del mare
- ✓ La precisione dipende dall'applicazione
 - Mappa per alpinisti oppure mappa per cartina stradale



Quantizzazione – 2

Data una grandezza che può assumere un qualsiasi valore continuo

- ✓ Si sceglie quanto approssimarla/arrotondarla
 - Si fa in modo che l'errore sia ridotto
 - Arrotondamento fine
 - Si deve misurare con precisione, si occupa molto spazio
 - Arrotondamento grossolano
 - Si rappresenta la grandezza in modo approssimativo
 - L'utente percepisce questo arrotondamento come un rumore aggiunto alla grandezza misurata



Quantizzazione – 3

In informatica l'arrotondamento si basa sulla quantizzazione

- ✓ I valori possibili sono numerati
 - Bisogna scegliere con cura cosa rappresentare
 - Valori non previsti non sono rappresentabili
- ✓ Ad ogni valore viene associato un numero intero
 - Simile alla codifica dei caratteri
- ✓ Una volta approssimato, il valore iniziale è perso
 - Quanti molto ravvicinati minimizzano la perdita
 - Compromesso tra precisione e costo (spazio)



Suoni – 1

Il suono consiste in oscillazioni nel tempo

- ✓ Il tempo è una dimensione continua
 - Campionamento nel tempo
 - Misurato in Hertz, ossia numero di campioni al secondo
- ✓ L'oscillazione varia con continuità
 - Quantizzazione dello scostamento dallo zero
 - Non ha un diretto corrispettivo fisico
 - Dipende dal volume massimo del suono rappresentato
- ✓ La scelta di campionamento e quantizzazione incide sulla qualità del suono



Suoni – 2

Valori tipici

- ✓ **Campionamento**
 - 8-10 kHz : voce per trasmissioni telefoniche
 - 44.1 kHz : musica a buona qualità (CD e DVD)
 - 96 kHz : musica in sala di registrazione o per conservazione
- ✓ **Quantizzazione**
 - 1 byte : voce a bassa qualità (poco usata)
 - 2 byte : musica a buona qualità (CD e DVD)
 - 3-4 byte : musica in sala di registrazione o per conservazione
- **La quantizzazione è particolarmente legata alla percezione della qualità sonora**



Immagini – 1

Le immagini sono distribuzioni di colori su un piano

- ✓ **La superficie è una dimensione continua**
 - **Campionamento nello spazio**
 - Misurato in dpi (dot per inch) ossia in punti per pollice
- ✓ **Il colore varia con continuità**
 - **Quantizzazione dell'insieme dei colori**
 - Diverse strategie possibili, in dipendenza dall'originale
 - Incide sul numero di diversi colori rappresentabili
- ✓ **La scelta di campionamento e quantizzazione incide sulla qualità dell'immagine**



Immagini – 2

Valori tipici

- ✓ **Campionamento**
 - 72-96 dpi : immagini a schermo o per il Web
 - 150-300 dpi : stampe su carta normale
 - 600 dpi e oltre : stampe su carta patinata
- ✓ **Quantizzazione**
 - 1 bit : bianco e nero
 - 4-8 bit : sfumature di grigio (da 16 a 256)
 - 1 byte : 256 colori diversi, scelti da una palette (tavolozza)
 - 3 byte : 16 milioni di colori, configurazione tipica
 - 4 byte : aggiunta di trasparenza o rappresentazione a 4 canali



Compressione

Definizione di compressione

- ✓ Conversione di un file in un secondo di minore dimensioni che porta la stessa informazione

Applicazioni

- ✓ **Riduzione dello spazio occupato in memoria**
 - Con riduzione dei tempi di caricamento dalla memoria di massa
- ✓ **Riduzione dei tempi di trasferimento**
- ✓ **Possibile riduzione dei tempi di calcolo**
 - A volte si può manipolare direttamente il file compresso
 - Se serve decomprimerlo i tempi di calcolo possono aumentare



Tipologie di compressione

Lossless (senza perdita)

- ✓ Il processo è completamente invertibile
 - Usata per testi, programmi, multimedia di qualità
 - Basata sulla ridondanza dell'informazione
- Esempi: archivi .zip, .rar, .tgz

Lossy (con perdita)

- ✓ Non si riesce a tornare perfettamente al file iniziale
 - Usata per immagini, suoni e video
 - Basata sulle soglie percettive umane
- Esempi: file .jpg, .mp3, .mpg, .avi



Compressione lossless

Considerazioni

- ✓ L'informazione può essere ridondante
 - Pixel di una zona monocroma di un'immagine
 - Parole ripetute, serie di spazi o a capo
- Si usano meno bit per descrivere le ridondanze
- ✓ Gli elementi hanno diverso contenuto informativo
 - Dopo la "q" è molto probabile trovare una "u"
 - Fase di attacco di un suono rispetto alla fase di nota tenuta
- Si usa un numero di bit proporzionale all'effettiva quantità di informazione



Compressione lossy

Considerazioni

- ✓ L'occhio e l'orecchio hanno dei limiti fisici
 - Ciò che non è percepito non è rappresentato
 - Dipende anche dalle interfacce di output
 - Si è effettuata un degrado al di sotto della soglia di percezione (di occhio e/o orecchio)
 - Attenzione: compressioni successive portano il degrado ad essere visibile/udibile (effetto fotocopia di fotocopia)
 - Scelta tra fattore di compressione e accettabilità
 - Si può accettare di percepire il degrado se è più importante avere file di piccole dimensioni