

Strutture Discrete

Informazioni generali e introduzione al corso
Docenti: Proff. Vincenzo Cutello e Claudia Cavallaro

Prof. V. Cutello

Contenuti

1 Informazioni generali

2 Introduzione al corso

- Il mondo discreto dell'Informatica
- Il mondo binario dell'Informatica
- Contenuti del Corso

Informazioni Generali

- Strutture Discrete è un corso di Informatica di 6CFU, corrispondenti a 48h di didattica frontale
- Non c'è un libro di testo specifico. Metterò su Teams (codice canale **y8nxa63**) i lucidi delle Lezioni (in pdf). Saranno più che sufficienti per rivedere i concetti presentati a lezione e prepararsi per gli esami.
- Quindi, sebbene la presenza non sia "obbligatoria" è invece fortemente consigliata sia in aula che online.

Prerequisiti ed esame

Come **prerequisiti per il Corso** è sufficiente avere buona conoscenza degli elementi di base dell'Aritmetica e dell'Algebra Elementare, ovvero le nozioni di base previste dai programmi di tutte le scuole medie superiori.

Piccolo dato statistico: gli studenti che frequentano e lavorano con costanza sui contenuti del corso, con alta probabilità superano l'esame entro i primi 2 appelli.

Contenuti

1 Informazioni generali

2 **Introduzione al corso**

- Il mondo discreto dell'Informatica
- Il mondo binario dell'Informatica
- Contenuti del Corso

Il significato di "Discreto"

- Prima di iniziare i nostri discorsi sulle "Strutture Discrete" chiariamone il significato.
- Un insieme (o una struttura di qualunque tipo) si dice "discreto" se
 - possiamo ordinare i suoi elementi in maniera tale che tra ogni elemento ed il successivo nell'ordine non vi sia nessun altro elemento.
 - Discreto:
 - L'opposto di "discreto" in matematica/informatica non è "indiscreto" ma "continuo": _____
- Esempi classici
 - Discreto: i numeri naturali \mathbb{N}
 - Continuo: i numeri reali \mathbb{R}

Informatica

- Il termine "informatica" è una contrazione di due termini: informatique = information + automatique.
- Quindi, il suo scopo è quello di progettare, costruire e far funzionare elaboratori/macchine per trattare in modo automatico l'informazione.
- Come? Per mezzo di Algoritmi. Un algoritmo è
 - una sequenza finita ed ordinata di operazioni
 - ognuna delle quali si completa in un tempo finito

Il mondo dell'informatica è "Discreto"

- RAM e HD strutture di memoria discrete
- il mondo "infinito" dei numeri viene delimitato, e per quanto riguarda i numeri reali
 - vengono campionati (un numero finito scelto) e
 - quantizzati (ai numeri con espansione decimale infinita è associata un'approssimazione)

Il mondo binario dell'Informatica

Per comunicare con il computer e farlo funzionare (Software), codificando l'informazione, serve un linguaggio formale. Quindi, come per tutti i linguaggi,

- Serve definire un Alfabeto, ovvero un elenco finito di simboli; e
- una grammatica formale, ovvero un insieme di regole sintattiche che specifichino come i simboli dell'alfabeto possano essere combinati tra di loro per costruire frasi ben formate all'interno del linguaggio.

Il mondo binario dell'Informatica

- La stessa informazione può essere rappresentata con codifiche diverse
 - Per esempio il numero 2023 può essere rappresentato anche usando la codifica romana con "MMXXIII"
- Un alfabeto minimale è quello binario, che è l'alfabeto di base dell'Informatica:
 - 2 soli simboli: 0 e 1 (bit = binary digit)
 - un byte è una sequenza di 8 bit (unità di misura base della capacità di memoria)

Sequenze binarie

Numero di sequenze con n bit

- 2 bit: 4 ($= 2^2$) "00, 01, 10, 11"
- 3 bit: 8 ($= 2^3$) "000, 001, 010, 011, 100, 101, 110, 111"
-
- 8 bit: 256 ($= 2^8$) "00000000, 00000001, ..., 11111110, 11111111"
- quindi in generale con n bits abbiamo 2^n sequenze possibili.

Bits and bytes

- La capacità di memoria (RAM e HD) è misurata in bytes
 - 1 KB (kilobyte) = 2^{10} bytes = 1.024 bytes
 - 1 MB (megabyte) = 2^{10} KB = 2^{20} bytes \simeq 1.000.000 bytes
 - 1 GB (gigabyte) = 2^{10} MB = 2^{20} KB = 2^{30} bytes \simeq 1.000.000.000 bytes
 - 1 TB (terabyte) = 2^{10} GB = 2^{20} MB = 2^{30} KB = 2^{40} bytes
- La capacità di trasmissione dati è misurata in bits
 - 1 Kb (kilobit) = 2^{10} bits
 - 1 Mb (megabit) = 2^{10} Kb = 2^{20} bits
 - 1 Gb (gigabit) = 2^{10} Mb = 2^{20} Kb = 2^{30} bits

Rappresentazione dei numeri naturali

Rappresentazioni non posizionali

- sistema di numerazione romano: additivo
- simboli:
 - $I = 1$
 - $V = 5$
 - $X = 10$
 - $L = 50$
 - $C = 100$
 - $D = 500$
 - $M = 1000$

Codice romano

Le cifre si sommano, ma una cifra più piccola a sinistra di una cifra più grande si sottrae.

- Esempi:
 - $II = 2$, $IV = 4$, $VIII = 8$
 - $XXX = 30$, $XXXIII = 33$, $XLIV = 44$
 - $XLIX = 49$, $LXXIV = 74$, $XCVIII = 98$

Rappresentazione posizionale dei numeri naturali

Rappresentazioni posizionali

- notazione decimale posizionale
- notazione posizionale in altra base
- notazione posizionale binaria
- notazione posizionale ottale
- notazione posizionale esadecimale

Codifica Posizionale

- Il codice decimale è un codice POSIZIONALE
 - $7 = 7 \times 10^0$ "Sette unità"
 - $75 = 7 \times 10^1 + 5 \times 10^0$ "Sette decine + cinque unità"
- Il valore di una cifra dipende dalla sua posizione e quindi non è assoluto.
- Ogni numero intero m si può scrivere come somma di potenze di 10, moltiplicate per una delle 10 cifre (da 0 a 9)

$$\sum_{i=0}^k b_i \cdot 10^i$$

Codifica Posizionale in base n

- Sia n un numero intero maggiore di 1.
- Abbiamo bisogno di n simboli diversi, per indicare tutti i valori interi (cifre) da 0 sino a $n - 1$.
- Ogni numero intero potrà essere codificato come somma di tali cifre da 0 a $n - 1$, moltiplicate per una potenza della base n .
- Esempi, oltre a $n = 10$
 - $n = 2$ (binario), cifre: 0, 1
 - $n = 8$, cifre: 0, 1, 2, 3, 4, 5, 6, 7
 - $n = 16$ (esadecimale), cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Codifica binaria dei naturali

- cifre (bit): 0, 1 e con n bit si possono rappresentare 2^n numeri naturali (da 0 a $2^n - 1$)
 - esempio:
$$10010101_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 149$$
- con un nibble (4 bit): 16 naturali da 0 a 15 (0000, 0001, 0010, ..., 1110, 1111). Notiamo che un nibble può essere rappresentato da una cifra esadecimale.
- Con un byte (8 bit): 256 naturali (00000000, ..., 11111111)

Codifica binaria dei naturali

- Il numero di bit necessari per codificare un numero $n > 1$ in binario è uguale ad $1 +$ la più grande potenza di 2 minore o uguale del numero n .
- Esempi:
 - $2, 2^1$, numero bits $1 + 1 = 2$
 - $3, 2^1$, numero bits $1 + 1 = 2$
 - $4, 2^2$, numero bits $1 + 2 = 3$
 - $5, 2^2$, numero bits $1 + 2 = 3$
 - $6, 2^2$, numero bits $1 + 2 = 3$
 - $7, 2^2$, numero bits $1 + 2 = 3$
 - $8, 2^3$, numero bits $1 + 3 = 4$

Computer come elaboratore di informazione

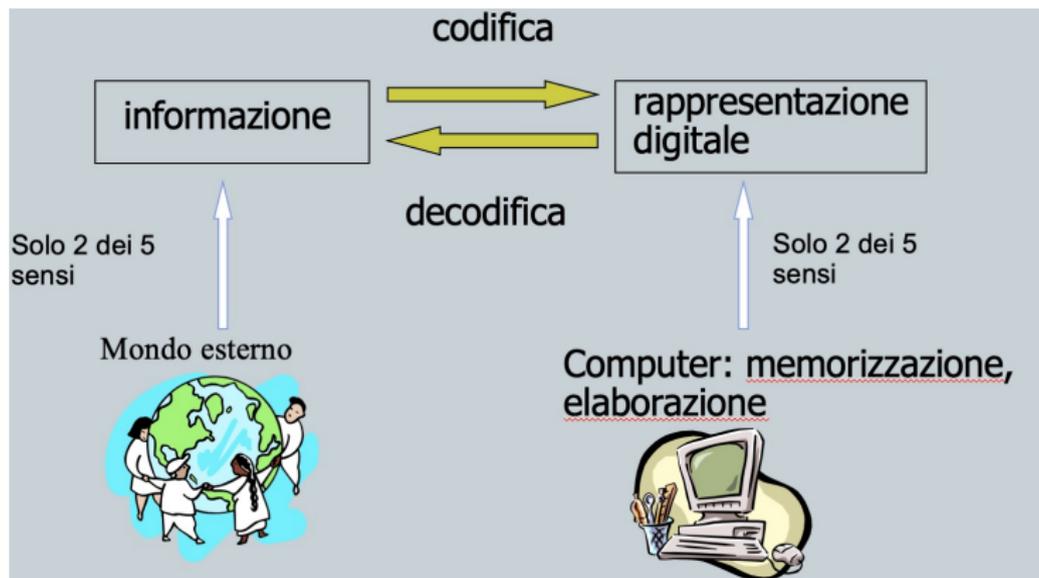


Figura: Dal mondo reale al computer

Computer come elaboratore di informazione

Un computer deve

- Fare input/output dell'informazione, usando i dispositivi di input/output
- Memorizzare l'informazione, usando la memoria principale/secondaria
- Elaborare l'informazione, usando il processore

Rappresentazione discreta

- La rappresentazione discreta è basata sulla combinazione di elementi minimi chiaramente distinti.
- Rappresentazione DIGITALE: è una rappresentazione discreta codificata tramite numeri
- Utilizzando la rappresentazione binaria, per esempio:
 - il mondo "continuo" delle immagini viene discretizzato/digitalizzato con un processo di campionamento e quantizzazione e rappresentato con pixels
 - i suoni vengono digitalizzati (campionamento e quantizzazione)
 - le mappe geografiche vengono discretizzate utilizzando strutture discrete (grafi)

Il mondo dell'informatica è "Discreto"

Le strutture discrete ed i concetti di matematica discreta sono alla base di quasi tutta l'informatica e le sue applicazioni. Per esempio

- Progettazione di reti ad alta velocità ed il routing dei messaggi
- Effettuare ricerche sul Web
- Progettare ed analizzare la complessità di algoritmi
- Progettare protocolli di crittografia

Matematica Discreta

In Matematica Discreta troviamo una serie di concetti e di tecniche di dimostrazione utilissime in molte applicazioni. Contenuti tipici sono

- 1 Logica
- 2 Insiemi e relazioni
- 3 Tecniche di dimostrazione ed induzione
- 4 Calcolo Combinatorio e Probabilità
- 5 Teoria dei numeri
- 6 Teoria dei grafi
- 7 Automi finiti e linguaggi

In questo corso ci occuperemo dei primi 6 punti, mentre l'ultimo ed ulteriori estensioni dei primi 6, saranno trattati in altri corsi

Contenuti del corso

Il corso, è suddiviso in 4 parti di uguale ampiezza, come sotto delineato. Ognuna delle parti si conclude con uno o più casi studio di particolare importanza.

- 1 Logica Proporzionale, Insiemi e relazioni (1,5 CFU)
- 2 Fondamenti di Teoria dei Numeri e metodologie di dimostrazione (1.5 CFU)
- 3 Calcolo Combinatorio e Probabilità Discrete (1.5 CFU)
- 4 Grafi (1.5 CFU)

Logica Proporzionale, Insiemi e relazioni

- Introduzione alla Logica Proporzionale e agli operatori di base.
- Il concetto di insieme e le proprietà di base. Insiemi ed operazioni tra di essi. Dimostrazione diretta.
- Famiglie di insiemi. Insieme prodotto. Paradossi.
- Relazioni binarie e funzioni. Relazioni di Equivalenza. Relazioni d'ordine, Rappresentazione di insiemi finiti.
- Il Problema del "Hitting Set"

Fondamenti di Teoria dei Numeri e metodologie di dimostrazione

- Numeri naturali, interi relativi, razionali.
- Divisione tra interi. Divisibilità e Criteri di divisibilità.
- MCD ed Algoritmo di Euclide. Numeri Primi e Coprimi.
- Aritmetica modulare. Congruenze. Proprietà delle congruenze. Invarianza rispetto a somma e prodotto
- Funzione ϕ di Eulero. Definizione e formula generale. Il Teorema di Eulero.
- Applicazioni dell'Aritmetica modulare.
- Teoria dei numeri e problemi aperti.
- Caso Studio: Il problema $3x + 1$ (Congettura di Collatz)

Calcolo Combinatorio e Probabilità Discrete

- Disposizioni e combinazioni. Permutazioni e Combinazioni.
- Teorema Binomiale. Il triangolo di Pascal. Combinazioni con ripetizione.
- Il Principio dei Cassetti (Pigeonhole Principle).
- Formalizzazione Matematica. Assiomi e Proprietà.
- La regola di Bayes. Problemi d'urna. Variabili casuali.
- Caso Studio : Giochi e paradossi probabilistici (Il paradosso di Monty Hall)

Introduzione alla Teoria dei Grafi

- Introduzione: strette di mano e passeggiate su ponti. Definizioni di base.
- Gradi di un nodo. Classi particolari di grafi: Grafi regolari, Grafi completi, Grafi Bipartiti.
- Sottografi, Isomorfismi e Omeomorfismi
- Percorsi, cammini e cicli. Grafi connessi. Rappresentazione di un grafo. Numero di percorsi tra nodi.
- Grafi Euleriani e grafi Hamiltoniani.
- Grafi pesati ed il problema del commesso viaggiatore.
- Colorazione di grafi e numero cromatico.
- Grafi planari, formula di Eulero e caratterizzazione della planarità.
- Definizione di Albero e caratterizzazione. Alberi binari e loro proprietà.
- Casi Studio: Esempi di problemi combinatori computazionalmente complessi.

Esame

Rispondiamo infine alla domanda che tutti gli studenti vogliono porre:

come sarà l'esame?

- Test scritto a risposte aperte. Due domande per ognuna delle 4 parti.

FINE INTRODUZIONE
