

Algoritmi e Principi dell'Informatica

Davide Martinenghi

Email: davide.martinenghi@polimi.it

<https://martinenghi.faculty.polimi.it/courses/api>

Ed 20, piano 1; tel: 3669

ricevimento

Venerdì 10:30 previo appuntamento via email

Struttura del corso

Articolato in due parti:

1. “Informatica teorica”
2. “Algoritmi e strutture dati”
 - Il primo modulo di API comprende i modelli dell’informatica (automi, logica) e la teoria della computazione (decidibilità...)
 - Il secondo modulo parte dallo studio della complessità per poi occuparsi di strutture dati ed algoritmi
 - Algoritmi presentati in *pseudo-codice*
 - Non c’è una parte sui linguaggi di programmazione

Prerequisiti

- Le basi essenziali di Informatica (Fondamenti di Informatica)
- Elementi di matematica non continua (Logica e Algebra)

Il materiale didattico

- Testi ufficiali:
 - Mandrioli/Spoletini: *Informatica teorica* (II edizione), De Agostini, 2011
 - Cormen et al.: *introduzione agli algoritmi e strutture dati* Terza edizione, McGraw-Hill
 - Versione «create» e e-book su VitalSource Bookshelf
 - Bertossi/Montresor: *Algoritmi e Strutture di Dati*, 3^a edizione. Città Studi Edizioni, 2014 (alternativo al precedente)
 - Barengghi/Martinenghi/Pradella/Rossi: *Algoritmi e Principi dell'Informatica: esercizi risolti e commentati*, 2^a edizione Esculapio, 2024
- Slide e temi d'esame risolti disponibili da subito sul sito

L'esame: tipologia

- Basato sulla *capacità di applicare, non di ripetere*:
 - Scritto con possibilità di consultare i testi ufficiali (e solo quelli!) e le slide (purché stampate)
 - Non è consentito l'uso di dispositivi elettronici!
 - L'orale non è previsto:
 - eventuale discussione orale esclusivamente su richiesta del docente

L'esame: appelli

- Tutti gli appelli sono divisi in due parti: 1) informatica teorica e 2) algoritmi e strutture dati. In ogni appello:
 - si può sempre partecipare a una sola parte o a entrambe;
 - se a entrambe, l'esame è superato se il totale è ≥ 18 , diviso in qualunque modo tra le due parti;
 - se l'esame complessivo non è superato, verrà comunque conservato il voto della parte sufficiente, cioè ≥ 9 , se c'è una parte sufficiente;
 - se a una sola, il voto parziale, se ≥ 9 , viene conservato per i 5 appelli di questa edizione del corso e sarà sommato al voto parziale ottenuto per l'altra parte (in precedenza o successivamente, nei 5 appelli di questa edizione del corso);
 - dopo ogni appello ci si può ritirare, anche da una sola parte.

L'esame: rifacimento e rifiuto

- Se si rifà una parte già valutata con un voto sufficiente (≥ 9), il voto precedente viene perso (basta partecipare all'esame per quella parte per considerarla rifatta, anche se poi ci si ritira o non si consegna)
 - Non si fa il max tra i due voti!
 - Esempio: ho 12/16 sulla prima parte, ma lo rifaccio e prendo 10/16? A questo punto, o tengo quest'ultimo parziale o dovrò rifarlo un'altra volta
- Se si rifiuta un voto dai Servizi Online, vengono persi entrambi i parziali
 - Ad esempio, se ho 12/16 sulla prima parte e 7/16 sulla seconda, se non accetto 19 dovrò rifare entrambe le parti

L'esame: insufficienze gravi

- Un compito gravemente insufficiente comporta la valutazione «riprovato»
 - Se pensate di essere gravemente insufficienti, non consegnate!
 - Le soluzioni saranno sul sito del corso subito dopo l'esame
 - Potrete comunque decidere di ritirarvi (entro il giorno dopo l'esame) dopo aver visto le soluzioni
- La soglia tipica per l'insufficienza grave è di 6 punti per l'esame complessivo e di 3 punti se si fa una singola parte
 - $6 \ll 18$, quindi dovete essere in grado di accorgervi se il vostro compito è lontano dalla sufficienza!

Prova finale

- Solo per chi l'ha nel piano di studi!
- Sarà presentata verso la fine di maggio e valutata:
 - Per i soli laureandi che ne facessero domanda, a luglio
 - Per tutti gli altri, in settembre
 - Per chi sarà laureando a febbraio dell'anno prossimo, si aprirà una finestra di valutazione in quel frangente
- Realizzazione di un progetto secondo specifiche date
- Valutazione automatica basata sul superamento di test con vincoli di efficienza... e **controlli anti-plagio molto severi**
 - NB: Il codice sviluppato non dev'essere reso pubblico (es., GitHub), se no gli altri possono copiarlo
 - Chi copia o fa copiare non supererà l'esame
 - Ogni anno capita di annullare qualche prova, anche a distanza di mesi

Difficoltà del corso

- Alcuni degli argomenti trattati nella parte di informatica teorica sono complessi
- Richiedono una capacità di astrazione e formalizzazione tipicamente non usata fino a questo punto degli studi
- Gli argomenti della seconda metà sono apparentemente più familiari, ma da non sottovalutare
- Il corso è complessivamente difficile!

Obiettivi e motivazioni

Perché l'informatica “teorica”?

La teoria stimolata dalla pratica:

generalità, rigore, “controllo”

- Comprendere a fondo e in maniera critica i principi dell'informatica (riesame approfondito di concetti elementari)
- Costruire basi solide per comprendere e dominare l'innovazione futura

Il programma

- **La modellizzazione informatica**
(Come descrivo un problema e la sua soluzione):
non tanto singoli modelli specialistici
quanto creare la capacità di capire e “inventare”
modelli vecchi e nuovi
- **La teoria della computazione:**
che cosa so/posso fare con lo strumento informatico
(quali problemi so risolvere)?

I modelli in campo ingegneristico

- In ingegneria la fase di progetto si basa sull'uso di *modelli*
- Infatti è spesso impossibile o impraticabile la verifica di soluzioni nel mondo reale
- Talvolta modelli *fisici* (es. ponte o diga in miniatura)
- Spesso modelli *formali*: oggetti matematici che fungono da rappresentazioni astratte di entità reali complesse

Uso dei modelli formali

1. Formalizzazione del problema: da entità reali ad astrazione matematica
2. Risoluzione del problema formalizzato
3. Interpretazione dei risultati ottenuti dal modello, ossia valutazione/deduzione delle scelte di progetto

Il modello è *adeguato* se i risultati ottenuti riflettono le proprietà che ci interessano del sistema fisico (entro limiti di approssimazione accettabili)

I modelli per l'informatica

Fasi principali dell'ingegneria del software classica:

1. Analisi dei requisiti → *documento di specifica*
2. Progetto → *architettura sw*
3. Implementazione → *codice*

Verso i modelli formali

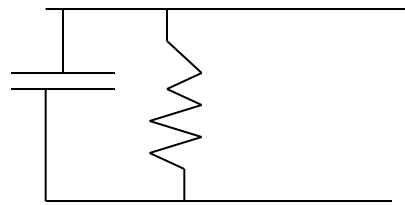
- Tradizionalmente si usa il linguaggio naturale
 - o semplici schemi con semantica informale
- La tendenza attuale è verso un uso sempre maggiore di linguaggi formali (o semi-formali) in tutte le fasi di produzione del software
 - Migliore affidabilità e facilità di manutenzione, ma soprattutto permette l'uso di **strumenti automatici**

I modelli dell'informatica: qualche nota

- Non (sol)tanto discreti anziché continui (bit e byte e non numeri reali), quanto:
 - *Generali*: il sistema informatico nel contesto (spesso) di un sistema più ampio: impianto, organizzazione, sistema “embedded”, ...
 - *Flessibili*: spesso non esiste il “modello già pronto”: occorre saper adattare modelli esistenti ad esigenze specifiche e impreviste
 - esistono molti (troppi) modelli specialistici: occorre saper studiare/inventare nuovi modelli

Differenza con i modelli di altre discipline

- Occorre (maggiore) attitudine dinamica e critica:
 - confronto modello-realtà
 - analisi e sintesi del modello/progetto
- Approccio diverso rispetto a:



oppure

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = g(x, y, z)$$

Formulazione del problema

- Spesso la vera difficoltà di un problema sta nel formularlo!

Che cosa significa:

- “automatizzare una procedura d’ufficio”
- “evitare incidenti ferroviari/aerei/...”
- ...

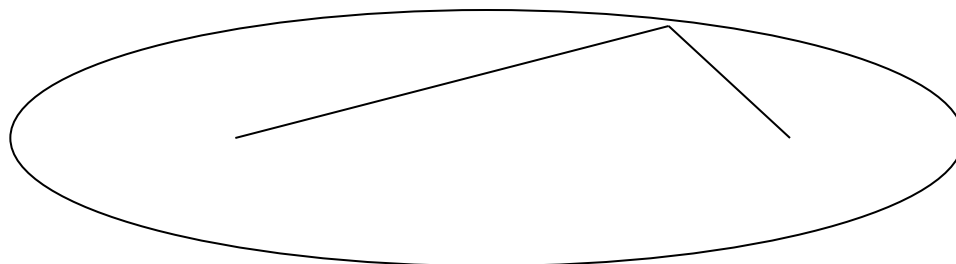
?

Tipi di modello

- Modelli *operazionali*
(macchine astratte, sistemi dinamici, ...)
basati sul concetto di stato e di meccanismi
(operazioni) per la sua evoluzione
- Modelli *descrittivi*
tesi a formulare le proprietà desiderate o temute del
sistema piuttosto che del suo funzionamento

Esempi di modello

- Modello operativo dell'ellisse:



- Modello descrittivo dell'ellisse:

$$ax^2 + by^2 = c$$

Altri esempi di modello

- Descrizione operativa dell'ordinamento:
 - Calcola il minimo e mettilo al primo posto;
 - Calcola il minimo degli elementi rimasti e mettilo al secondo posto;
 - ...
- Descrizione non-operativa dell'ordinamento:
 - Individua una permutazione della sequenza data tale che $\forall i (a[i] \leq a[i+1])$

Conclusioni

- In realtà le differenze tra modellizzazione operativa e modellizzazione descrittiva non sono così nette: più che altro si tratta di un utile riferimento nel classificare un tipo di modello