

Introduzione all'Informatica

Davide Bresolin
a.a 2026/27



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

1 Che cos'è l'informatica

2 Pensare da informatici

Che cos'è l'informatica?



L'informatica è uno strumento per **descrivere** e **risolvere**
problemi

Che cos'è l'informatica?



L'informatica è uno strumento per **descrivere** e **risolvere problemi** mediante **una serie** di “calcoli”

Ingredienti

- Uova (1 tuorlo a testa più un uovo intero ogni 4 persone)
- Pecorino o Parmigiano (35 grammi a persona)
- Guanciale o Pancetta (40 grammi a persona)
- Pasta (100 grammi a persona)
- Sale e Pepe

Procedura

- 1 Cuocere la pasta in acqua salata
- 2 Sbattere le uova con pecorino e pepe
- 3 Rosolare il guanciale a cubetti o striscioline a fuoco medio
- 4 Scolare la pasta al dente
- 5 Terminare la cottura della pasta nella padella dove state rosolando il guanciale
- 6 Spegner il fuoco. Quando si raggiunge la temperatura di 65°C , aggiungere l'uovo sbattuto con il pecorino

Algoritmo

insieme di passi che definiscono il modo in cui viene risolto un problema

- il risultato che si vuole ottenere deve essere definito in modo preciso e verificabile
- gli input del problema devono essere manipolabili da un computer
- i passi da svolgere devono essere eseguibili da un computer

- Prima che un computer possa risolvere un problema è necessario **scoprire un algoritmo** che lo risolve
- L'algoritmo deve essere **rappresentato in una forma compatibile con la macchina**
- La rappresentazione di un algoritmo si chiama **programma**

- Per comodità degli esseri umani, i programmi sono **stampati** su carta o **visualizzati** su uno schermo
- Per la comodità delle macchine, i programmi sono **codificati in modo compatibile con la macchina**
- Il processo di sviluppo di un programma e la sua codifica in forma compatibile con la macchina si chiama **programmazione**.

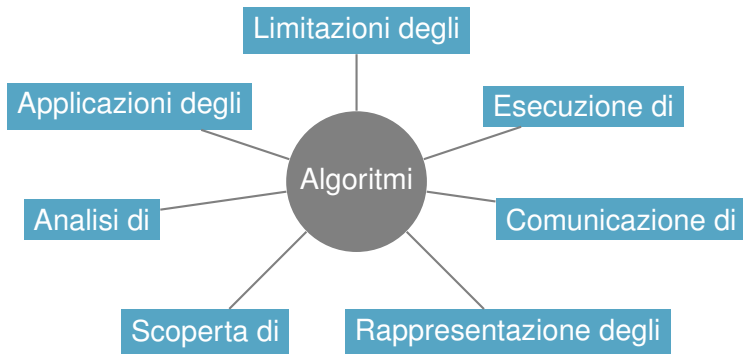
- I programmi e gli algoritmi che essi rappresentano sono definiti **software**:
 - Descrive le operazioni che **trasformano informazione**
 - È **facilmente modificabile dall'utente**
- La macchina stessa è nota come **hardware**:
 - Descrive la **struttura fisica** del calcolatore: le operazioni che è in grado di eseguire e come comunica l'informazione con l'esterno
 - **Non è modificabile dall'utente**

- lo studio degli algoritmi è nato come questione matematica
- la ricerca di algoritmi è iniziata molto prima dello sviluppo dei computer
- **obiettivo:** trovare un unico insieme di operazioni che permettono di risolvere tutti i problemi di un particolare tipo
- Una volta trovato un algoritmo, risolvere il problema si riduce al semplice processo di seguire le indicazioni dell'algoritmo

- Si può dire che l'intelligenza necessaria per risolvere il problema è codificata nell'algoritmo
- Catturare l'intelligenza attraverso gli algoritmi ci permette di costruire macchine che eseguono compiti utili.
- Il livello di intelligenza delle macchine è limitato dall'intelligenza che può essere trasmessa attraverso gli algoritmi

- Possiamo costruire una macchina che svolga un compito solo se esiste un algoritmo per eseguirlo.
- Viceversa, se non esiste un algoritmo per risolvere un problema, la soluzione di quel problema è al di là delle capacità delle macchine
- L'identificazione dei limiti delle capacità algoritmiche si è consolidata come argomento in matematica negli anni Trenta
- lo studio delle capacità algoritmiche che ne derivò fu l'inizio del campo oggi noto come informatica.

È lo studio degli algoritmi a costituire il nucleo dell'informatica.



- 1 il modo in cui le informazioni vengono codificate e memorizzate all'interno dei computer
- 2 la rappresentazione degli algoritmi e il processo di sviluppo di programmi in Python
- 3 chiudiamo il nostro studio con un paio di esempi di algoritmi applicati alla Biologia

1 Che cos'è l'informatica

2 Pensare da informatici

Come i matematici...

...gli informatici si avvalgono di **linguaggi formali** per esprimere dei concetti

Come gli ingegneri...

...gli informatici **progettano** svariate cose, compongono singole parti in sistemi e valutano costi e benefici tra varie alternative

Come i naturalisti...

...gli informatici **osservano** come si comportano dei sistemi complessi, formulano delle ipotesi e verificano le loro previsioni



Risolvere problemi richiede:

- formulare il problema
- pensare in maniera creativa alle possibili soluzioni
- esprimere una soluzione in modo chiaro ed accurato

Imparare a programmare è un ottimo modo per mettere alla prova la propria bravura nel risolvere problemi!

Un **programma** è una serie di **istruzioni** che spiegano come effettuare un “calcolo”:

- un calcolo di tipo **matematico**
 - soluzione di equazioni, radici di un polinomio, . . .
- un calcolo di tipo **simbolico**
 - ricerca di una parola in un testo, manipolazione di sequenze di DNA o RNA
- un calcolo di tipo **multimediale**
 - elaborazione di un'immagine, visualizzazione di un filmato

Ci sono cinque tipi di istruzioni fondamentali:

input: ricevimento di dati dalla tastiera, da un file, dalla rete, ...

output: invio di dati allo schermo, salvataggio su un file, trasmissione verso la rete, ...

matematiche: esecuzione di operazioni matematiche di base, come addizione, moltiplicazione, ...

condizionali: controllo di determinate condizioni ed esecuzione del blocco di istruzioni appropriato

ripetizioni: esecuzione ripetuta di una certa azione

Linguaggi naturali

- sono le lingue parlate, come italiano, inglese, francese
- non sono opera di qualcuno, si sono **evoluti naturalmente**

Linguaggi formali

- sono appositamente **creati per specifiche applicazioni**
- la notazione matematica, la notazione delle reazioni chimiche, . . .

Linguaggi di programmazione

Linguaggi formali progettati per descrivere algoritmi.

- I linguaggi formali seguono **regole rigide** che vanno rispettate alla lettera
- **Regole sui simboli:** definiscono gli elementi di base
 - $3 + 3 = \$6$ non è una espressione valida perché \$ non è un simbolo consentito
 - Zz_2 non è una formula chimica corretta, perché Zz non è un elemento chimico

- I linguaggi formali seguono **regole rigide** che vanno rispettate alla lettera
- **Regole sulla struttura:** riguardano il modo in cui sono disposti i simboli
 - $3+ = 3$ non è strutturalmente corretta, perché non si può mettere $=$ dopo $+$
 - ${}_2Pb$ non è strutturalmente corretta, perché non si può mettere il pedice prima del simbolo chimico

Linguaggi naturali vs. linguaggi formali



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

	Linguaggi naturali	Linguaggi formali
Ambiguità	✓	
Ridondanza	✓	
Letteralità		✓



Un linguaggio di programmazione . . .

- è **denso di significato**: occorre più tempo per leggerlo
- la **struttura è importante**: l'ordine in cui vanno letti non è sempre quello dall'alto in basso e da sinistra a destra
- i **dettagli contano molto**: errori di ortografia e punteggiatura possono diventare molto gravi

Pensate al computer come ad un **impiegato** con **punti di forza**:

- velocità
- precisione

e **debolezze**:

- mancanza di empatia
- incapacità di cogliere il quadro generale

Il vostro compito sarà di **essere un buon manager**:

- trarre vantaggio dai pregi
- mitigare i difetti

I programmatori commettono errori!



- gli errori di programmazione sono detti **bug**
- il procedimento della loro ricerca e correzione è chiamato **debug**
- ci sono **tre tipi di errori**:
 - **Errori di sintassi**: il programma non segue le regole sui simboli o sulla struttura
 - **Errori in esecuzione**: interrompono l'esecuzione del programma
 - **Errori di semantica** (o di logica): il programma esegue senza errori, ma non fa' la cosa giusta
- identificare il tipo di errore è il primo passo per correggerlo!

