

# *Introduzione al pensiero computazionale*

corso di laurea in **Matematica**

*Informatica Generale*, Lezione **1(a)**

**Ivano Salvo**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

# Cos'è l'Informatica?

Nel 2022, il **Informatica Generale** come nome di un corso è quantomeno **pretenzioso**.

L'informatica nasce dalla necessità di **calcolare in modo meccanico**, ma oggi è **pervasiva** in ogni attività umana.

☞ L'**Informatica** si occupa del **trattamento** e **comunicazione** dell'**informazione** in modo **automatico**.

Come tale oggi abbraccia un **gran numero di aree**, come:

- ▶ calcolatori elettronici (architetture) e sistemi operativi
- ▶ reti di calcolatori, sicurezza e crittografia
- ▶ basi di dati e Big Data
- ▶ intelligenza artificiale
- ▶ **algoritmi**
- ▶ **programmazione** e correttezza del software
- ▶ ... molto altro!

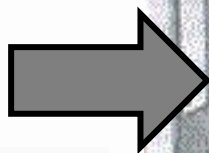
*affronteremo solo questi  
due fondamenti del  
pensiero computazionale  
che **sono ingredienti** di  
tutte le altre discipline*

# *Un ricordo del prof. Corrado Böhm*



*“L'informatica è al tempo stesso  
**madre** e **figlia** della Matematica”*

*inaugurazione del primo  
calcolatore a Roma (1955)  
(secondo in Italia) allo  
IAC: ci sono, tra gli altri,  
**Mauro Picone**, Roberto  
Vacca, Giorgio Sacerdoti*



# *Esempi di Matematica come calcolo*

**Teorema di Pitagora** e criteri di similitudine/uguaglianza dei triangoli...

◆ permettono di calcolare una grandezza incognita conoscendone altre da cui dipende

costruzioni con **riga e compasso**...

◆ offrono delle "procedure" per costruzioni geometriche

**logaritmi**...

◆ facilitano i calcoli trasformando prodotti in somme

proprietà algebriche e **rappresentazioni posizionali** dei numeri...

◆ portano ad algoritmi efficienti e meccanici (=basati su manipolazioni di simboli) per somma/prodotto/divisione di numeri a più cifre

...

# Matematica “non Informatica”

**Teorema:** esistono due numeri irrazionali  $x$  e  $y$  tali che  $x^y$  è razionale.

**Dimostrazione:** Sia  $x=\sqrt{2}$  e  $y=\sqrt{2}$ . Se  $\sqrt{2}^{\sqrt{2}}$  è razionale ho la tesi.  
Altrimenti, pongo  $x = \sqrt{2}^{\sqrt{2}}$ . Ora  $x^y = \sqrt{2}^{\sqrt{2}\sqrt{2}} = \sqrt{2}^2 = 2$ .

*il teorema non  
dice quali siano i  
due numeri!*

**Teorema:** [BOLZANO-WEIRSTRASS] Un insieme infinito di punti  $I$  in un intervallo chiuso e limitato  $[a, b]$  ha un punto di accumulazione.

**Dimostrazione:** Sia  $c_0$  il punto medio di  $[a, b]$ : almeno uno dei due intervalli  $[a, c_0]$  e  $[c_0, b]$  conterrà un numero infinito di punti di  $I$ .

Prendo tale intervallo e prendo il punto medio  $c_1$ . Almeno uno dei due sotto-intervalli conterrà ancora un numero infinito di punti di  $I$ .

Costruisco una successione di intervalli (inclusi) e di punti  $\{c_k\}_{k \in \mathbb{N}}$  di dimensione  $\frac{b-a}{2^k}$ . Detto  $c = \lim_{k \rightarrow \infty} c_k$  ho che ogni intorno di  $c$  contiene infiniti punti di  $I$ . □

*Non abbiamo un  
metodo per  
calcolare il limite*

# Il Problema del mercato

Siete stati "invitati" a **risolvere problemi** fin dall'infanzia. Ad esempio: Geppetto va al mercato, compra **3 pere** per Pinocchio ogni pera costa **14 soldi**. Quanto spende Geppetto?



In questo caso, la soluzione è semplicemente un **numero**.

La risposta, ovviamente, è **42 soldi**.

Notate che i personaggi siano Geppetto, pinocchio e 3 pere sono **dettagli non rilevanti** per calcolare la soluzione. **Astrazione!**



# *Problema: salvare capra e cavolo*

Un **barcaiolo** deve trasportare da una sponda all'altra di un fiume un **lupo**, una **capra** e dei **cavoli** su una barchetta.

La barca **non può trasportare più di una cosa insieme.**

Trovare l'esatto ordine di azioni affinché il **lupo non mangi la capra** o la **capra non mangi i cavoli.**



In questo caso, la soluzione è una **sequenza di azioni.**

**Esercizio.**

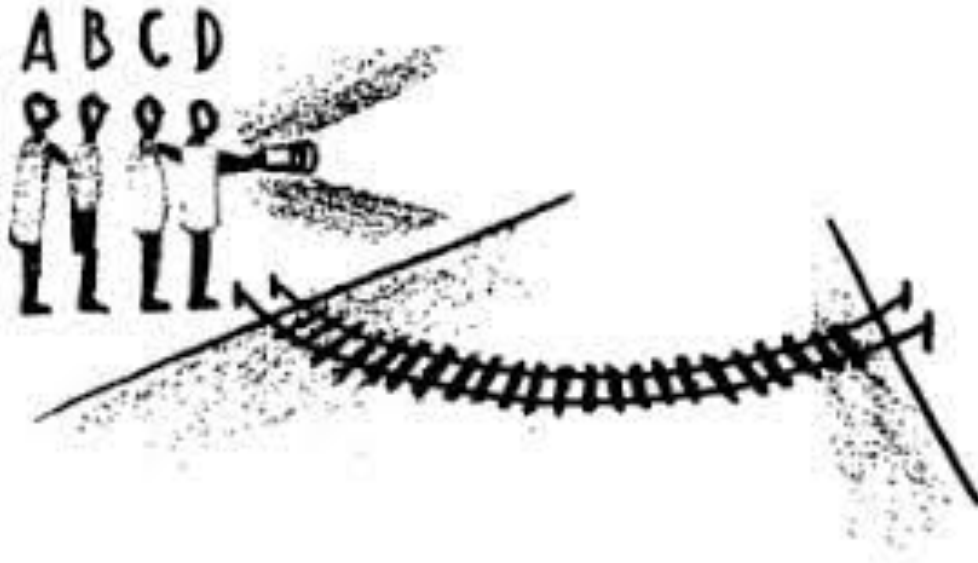
# *Attraversare il ponte con una torcia*

Quattro persone devono attraversare un ponte al buio.

C'è **un'unica torcia**. Il ponte può sostenere al **massimo due persone**.

Ognuno ha tempi diversi: **A** 10min, **B** 5min, **C** 2min, **D** 1 min:  
quando passano in due, siccome devono stare vicini, il **tempo di attraversamento** è quello della **persona più lenta**.

Trovare la sequenza che **minimizza** il tempo totale.



In questo caso, la soluzione è una **sequenza di azioni** che **ottimizza** il tempo di attraversamento.

**Esercizio.**

# *Il Problema del mercato revisited*

Il signor Tal de' Tali va al mercato. Compra  $n$  oggetti al prezzo  $p$ .  
Quanto spende il signor Tal de' Tali?



In questo caso, la risposta è una procedura risolutiva:

*“**esegui la moltiplicazione**  
 $n \times p$ ”*

che risolve una intera **classe di problemi**.

Ogni istanza si differenzia per i valori di  $n$  e  $p$ .

# Somma "unaria"

In realtà il primo metodo che avete imparato per sommare due numeri consisteva nel **contare sulle dita**.

Immaginiamo di eseguirlo, usando un foglio di carta al posto delle dita...

|     |                                          |
|-----|------------------------------------------|
| $m$ | <del>5</del> <del>6</del> <del>7</del> 8 |
| $n$ | <del>3</del> <del>2</del> <del>1</del> 0 |

Per **descrivere** l'algoritmo, ci conviene dare un nome alle due "caselle di memoria"

*modifica della  
memoria  
(distruttiva)*

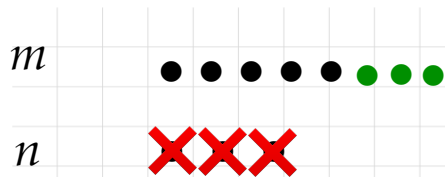
"Cancella il contenuto di  $m$  e sostituiscilo con  $m+1$  e cancella il contenuto di  $n$  e sostituiscilo con  $n-1$  **finché** il contenuto di  $n$  non diventa 0"

*ripetizione di  
azioni*



# Somma “unaria” in notazione unaria

Questo algoritmo (circa uguale) funziona anche se rappresentiamo i numeri in **notazione unaria**.



“Cancella un pallino dalla rappresentazione di  $n$  e scrivilo nella rappresentazione di  $m$ , finché ci sono pallini nella rappresentazione di  $n$ .”

*il risultato, in questo caso, è  
l'**ultima parola** che si pronuncia  
mentre si disegnano i pallini di  
una **filastrocca ben nota**:  
“cinque”, “sei”, “sette”, “**otto**”*



# Somma di due numeri a più cifre

Problema: calcolare la somma di 1621 e 746

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| +   | 0   | 1   | ... | 6   | ... | 9   |
| 0   | 0   | 1   | ... | 6   | ... | 9   |
| 1   | 1   | 2   | ... | 7   | ... | 0   |
| ... | ... | ... | ... | ... | ... | ... |
| 7   | 7   | 8   | ... | 3   | ... | 6   |
| ... | ... | ... | ... | ... | ... | ... |
| 9   | 0   | ... | ... | 5   | ... | 8   |

conoscenza necessaria

|     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|
| R   | 0   | 1   | ... | 6   | ... | 9   |
| 0   | 0   | 0   | ... | 0   | ... | 0   |
| 1   | 0   | 0   | ... | 0   | ... | 1   |
| ... | ... | ... | ... | ... | ... | ... |
| 7   | 0   | 0   | ... | 1   | ... | 1   |
| ... | ... | ... | ... | ... | ... | ... |
| 9   | 0   | 1   | ... | 1   | ... | 1   |

$$\begin{array}{r} 1621 \\ + 0746 \\ \hline 2367 \end{array}$$

memoria necessaria durante il calcolo

Nell'esecuzione del calcolo potete **ignorare il significato** dei simboli. Basta conoscere **tabella di somme e riporti**.

# *Detour storico*

La parola **algoritmo** deriva probabilmente dal nome del matematico persiano **Muḥammad ibn Mūsā al-Khwārizmī** (~780-~850), a cui sono attribuiti gli algoritmi per le operazioni aritmetiche sui numeri rappresentati in forma posizionale.



# Descrizione dell'algoritmo di somma

È necessario **descrivere un algoritmo** in un **linguaggio univocamente interpretabile** dall'**esecutore**.

1. scrivere i due numeri, **allineandoli a destra**
2. cominciando da **destra verso sinistra**, considerare le due cifre  $c$  e  $c'$  cifre che si trovano nella **stessa colonna**
3. Scrivere il numeretto che si trova nella tabella  $+$ , in corrispondenza della riga  $c$  e colonna  $c'$
4. Scrivere il numeretto che si trova nella tabella  $R$ , in corrispondenza della riga  $c$  e colonna  $c'$  sopra le cifre nella colonna immediatamente a sinistra
5. **Ripetere** [le operazioni 2-4] finché **non sono esaurite** le cifre e i riporti

*Usualmente per le operazioni 3/4  
si assume l'esecutore capace di **sommare**  
**due numeri di una cifra***

# Rappresentazione dei Dati

L'algoritmo appena descritto è **intimamente collegato** con la **rappresentazione posizionale** (in base 10) dei numeri naturali.

Non sarebbe possibile eseguirlo se i numeri fossero, ad esempio in notazione unaria! o numeri romani!

La sua **correttezza** dipende da alcune **proprietà algebriche** di somma e prodotto, sfruttate da questa rappresentazione.

La **rappresentazione posizionale** dei numeri a base  $b$  deve la sua fortuna a tre proprietà:

- ▶ usa solo  **$b$  simboli distinti** (sempre gli stessi)
- ▶ è **economica** (servono  **$\log_b n$  simboli** per rappresentare  $n$ )
- ▶ esistono **algoritmi facili** ed “**efficienti**” per fare le operazioni.

# *Algoritmo di somma: osservazioni*

- ▶ Si può eseguire **senza sapere il significato** di ciò che si fa purché l'esecutore sia in grado di **consultare** (o sapere a memoria) le tabelle di **somme e riporti**.
- ▶ Dipende **non tanto dai numeri naturali**, ma è legato a una **particolare rappresentazione** (o **codifica**) dei numeri naturali e alle sue proprietà (fondate sulla proprietà distributiva...)
- ▶ La descrizione funziona per numeri **arbitrariamente lunghi** (**illimitati**) ma ciascuna esecuzione opererà su numeri di **lunghezza finita**.
- ▶ L'algoritmo è relativo a un **esecutore** con alcune **capacità elementari**: **allineare** i numeri a **destra**, **consultare** le tabelle, **scrivere/leggere/cancellare/riscrivere** cifre sul foglio di carta (**memoria**) o capire che il calcolo sia finito.
- ▶ L'esecutore deve essere in grado di **ripetere** alcune operazioni **fino al verificarsi** di precise **condizioni**.

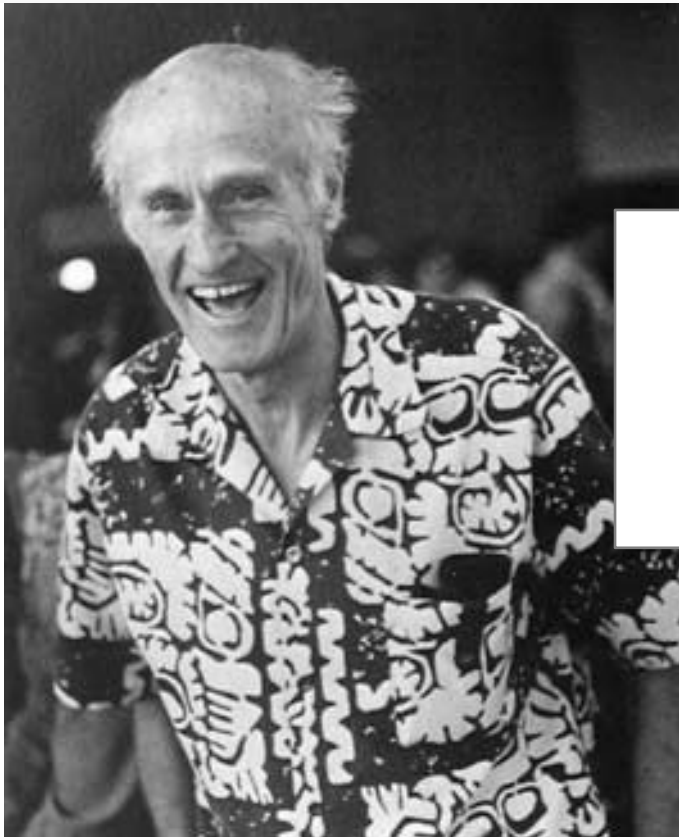
# *Caratteristiche di un Algoritmo*

- ▶ Le singole **azioni elementari** devono essere interpretabili in **modo univoco** dall'esecutore...
- ▶ l'effetto di ciascuna azione elementare è **univocamente determinabile** (**deterministico**)
- ▶ l'esecutore opera in **modo discreto**
- ▶ La **descrizione** di un algoritmo è **finita**...  
... ma arbitrariamente lunga
- ▶ Un algoritmo opera su dati di **dimensioni illimitate**...  
... ma ogni esecuzione accetta un **input finito**.
- ▶ Un algoritmo usa una **memoria illimitata**...  
... ma ogni esecuzione usa **memoria finita**.
- ▶ Un algoritmo fa un **numero di operazioni illimitato**...  
... ma ogni esecuzione termina in **tempo finito**.

# *Fuori Programma*

Questa lezione ha l'obiettivo di circoscrivere cosa sia un algoritmo, e quali siano le sue caratteristiche.

Sfogliando un libro di complessità computazionale, uno dei capitoli si apriva con questa citazione folgorante:



An algorithm is a finite answer to an infinite number of questions.

Stephen Kleene

# *La ricetta è un algoritmo?*

Prendete la ricetta del vostro piatto preferito...

Cercate di descriverla nel modo più accurato possibile...

Ma è un **algoritmo**?

Mostrare che **viola diverse caratteristiche** viste prima!



Eppure, Chiara Valerio, dovendo spiegare cosa sia un algoritmo...

...fa proprio l'esempio di una ricetta.

Che esempi avrebbe potuto fare?

# *I calcolatori pensano? E gli umani?*

Le caratteristiche di un algoritmo implica che siamo interessati a **esecutori** (= calcolatori elettronici) capaci di **complesse manipolazioni simboliche** in **pochissimo tempo**, **senza pensare al significato** dei calcoli che eseguono.

Tuttavia, i calcolatori sembrano **simulare pensiero intelligente**.



Ma forse neanche noi esseri umani pensiamo veramente :/