



**UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE**

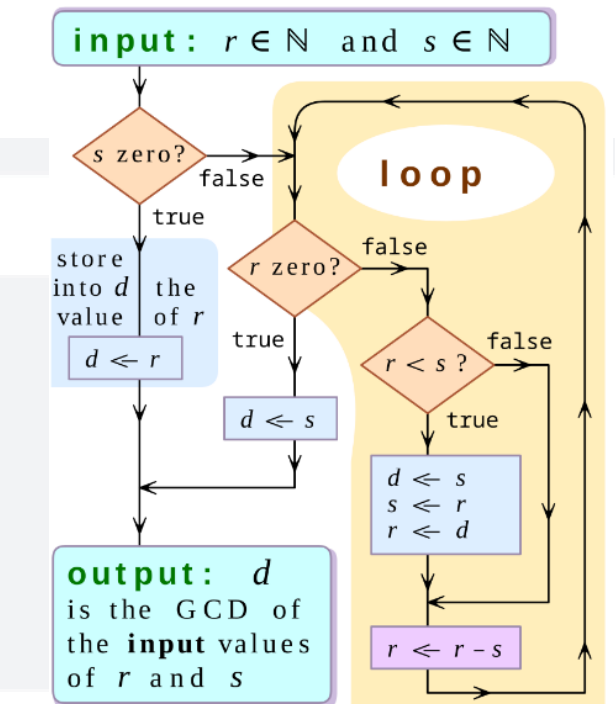
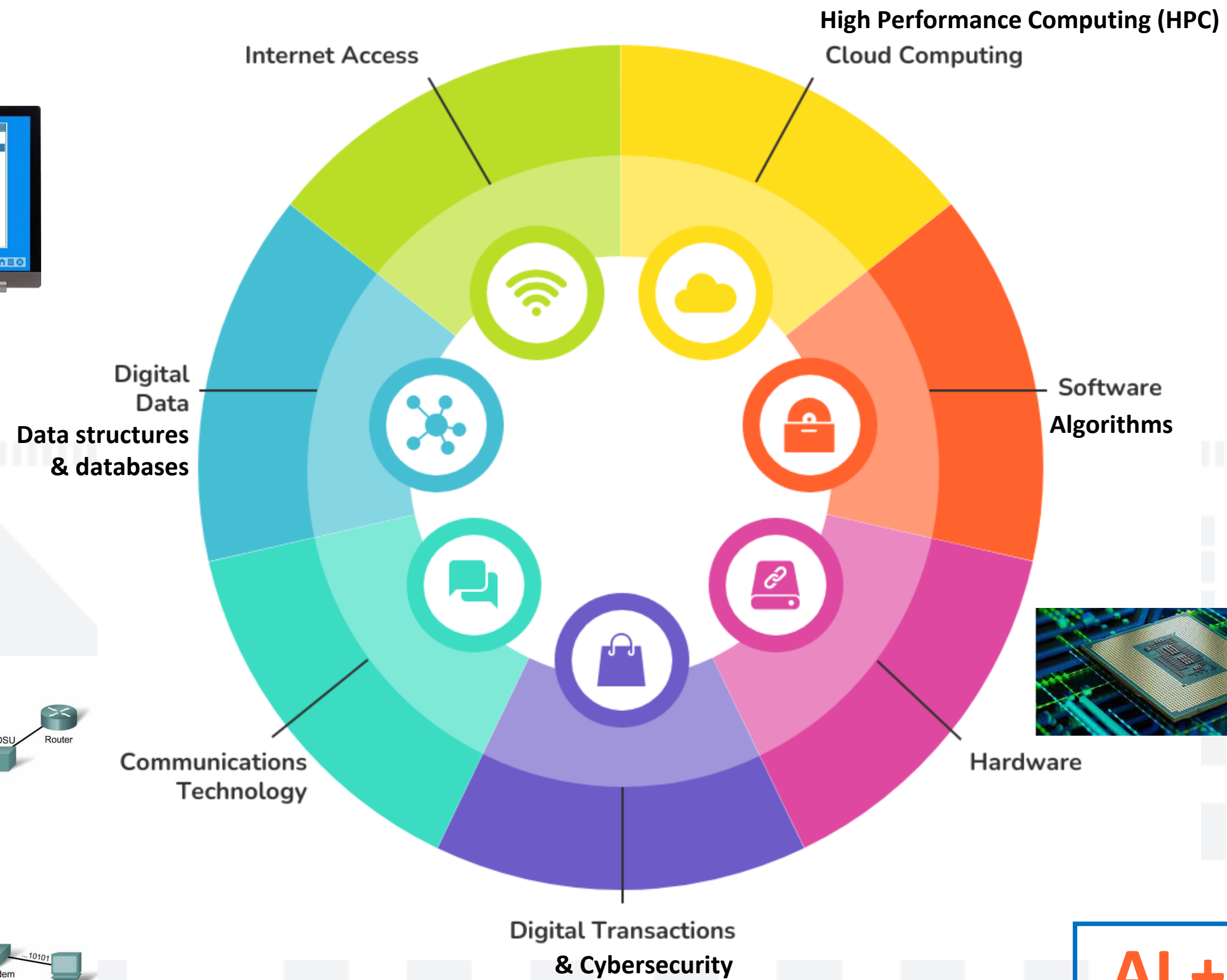
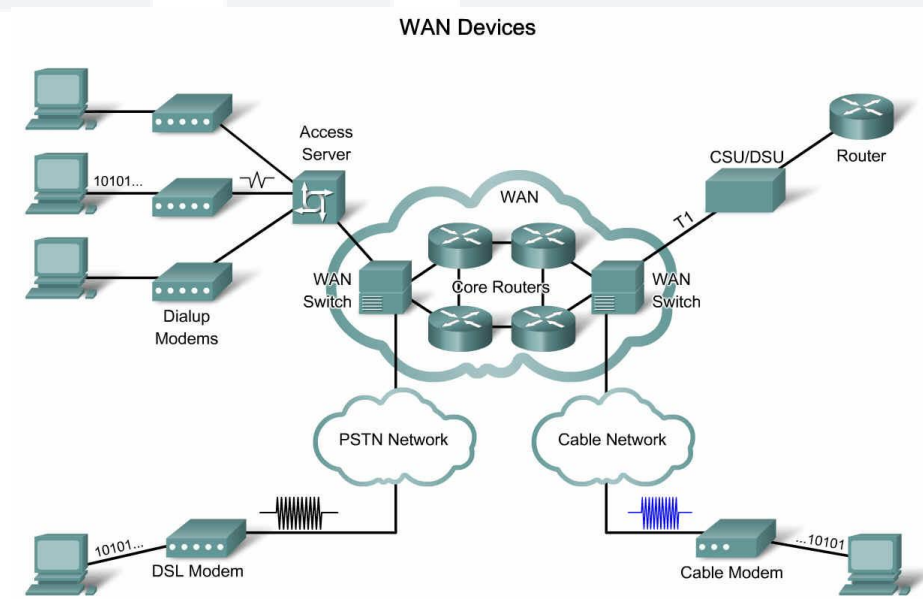
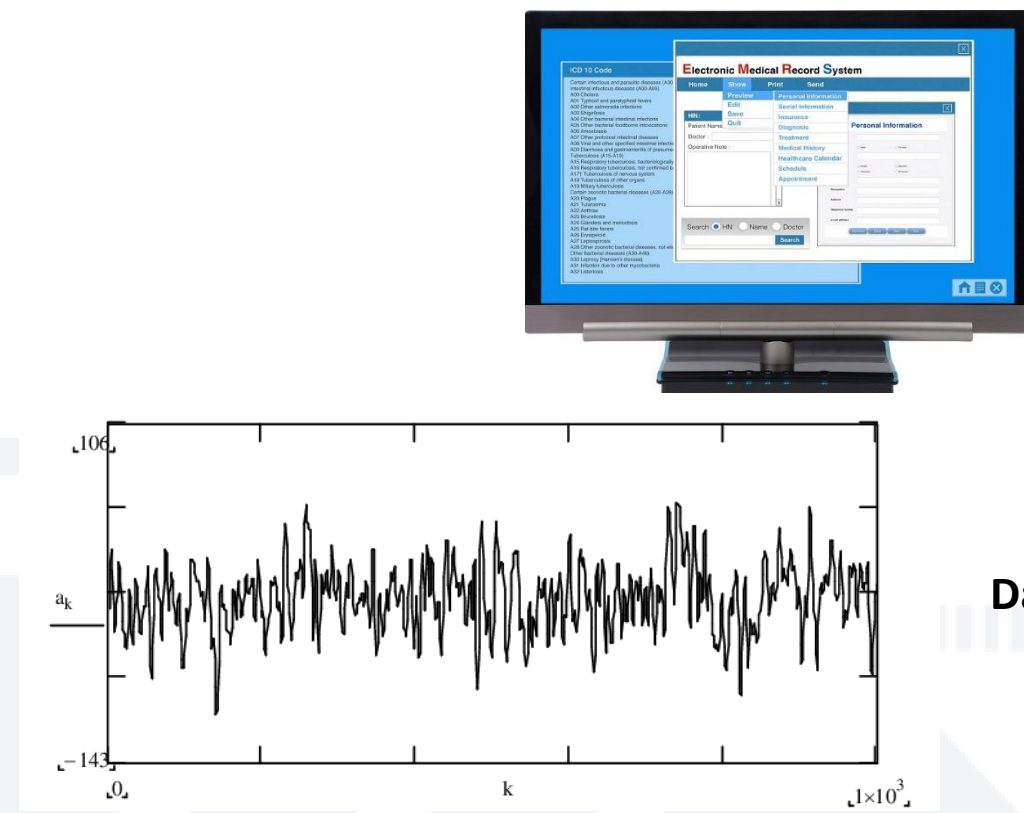
# Corso di INFORMATICA: *Considerazioni finali*

Ed. 2025/2026

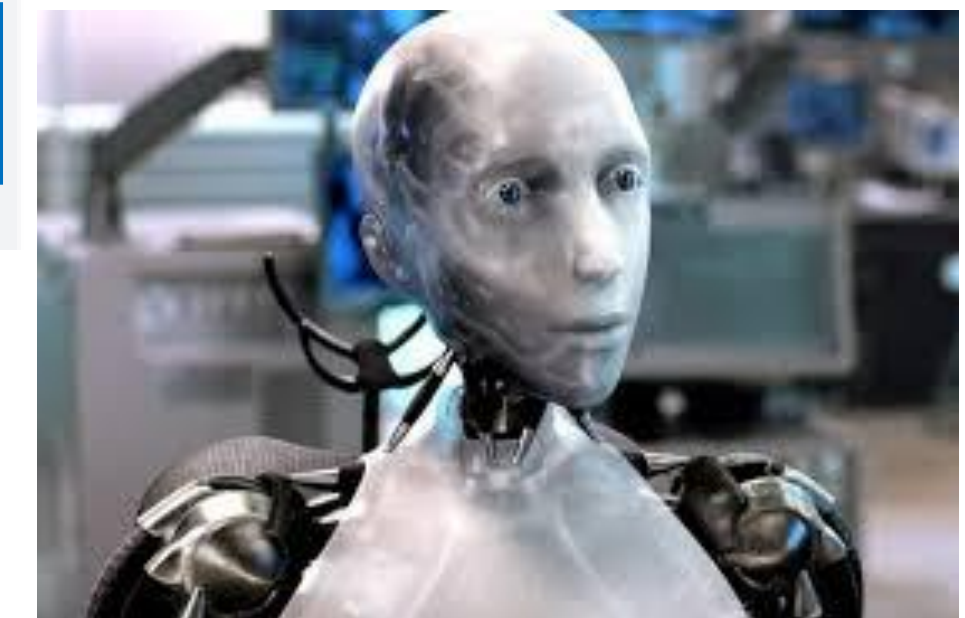
**Prof.ssa Giulia Cisotto**

Trieste, 14 maggio 2026

# INFORMATICA



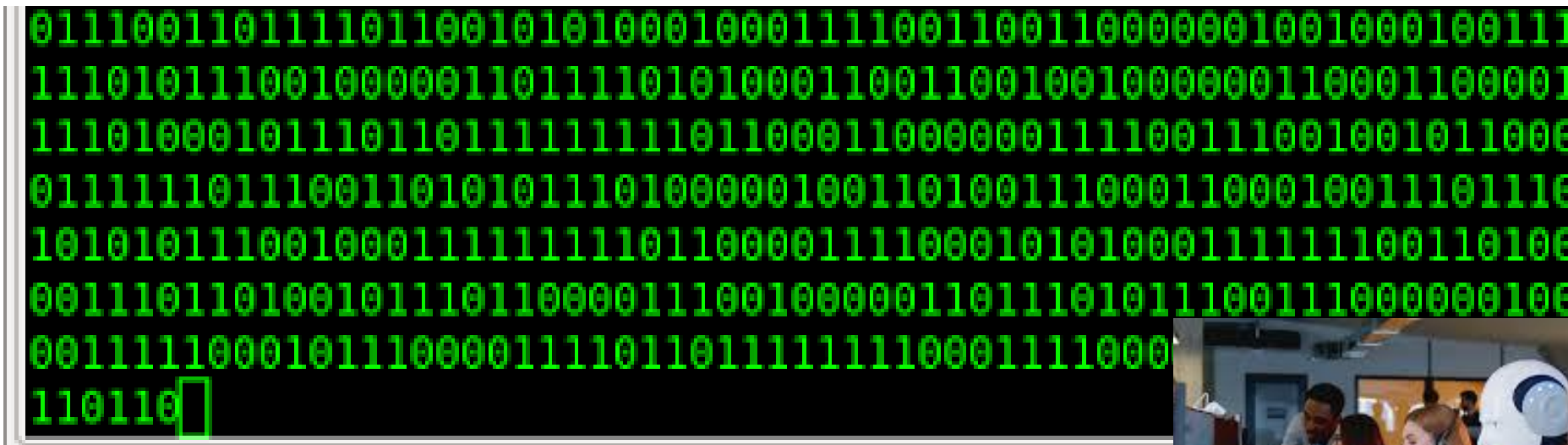
**AI + robotica**



# Che problemi affronta l'informatica?

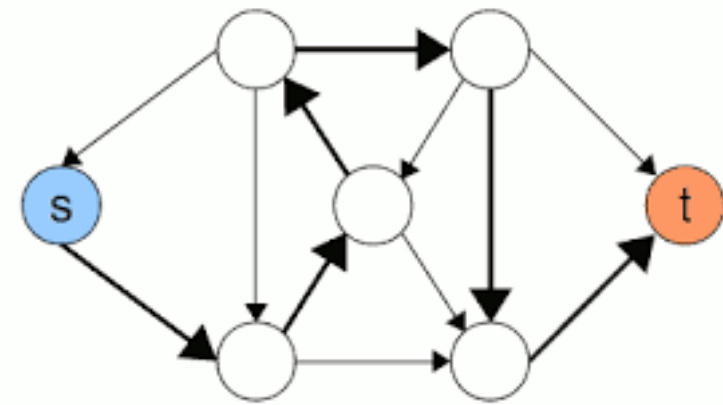
Come **COMUNICARE**  
con i computer e le macchine

Rappresentazione  
dell'informazione

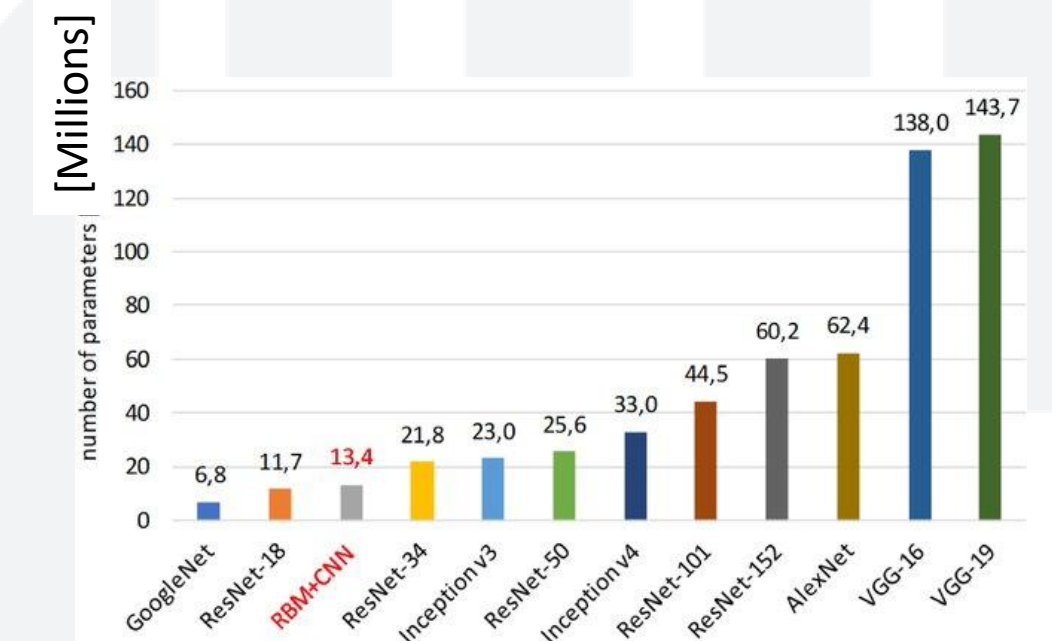
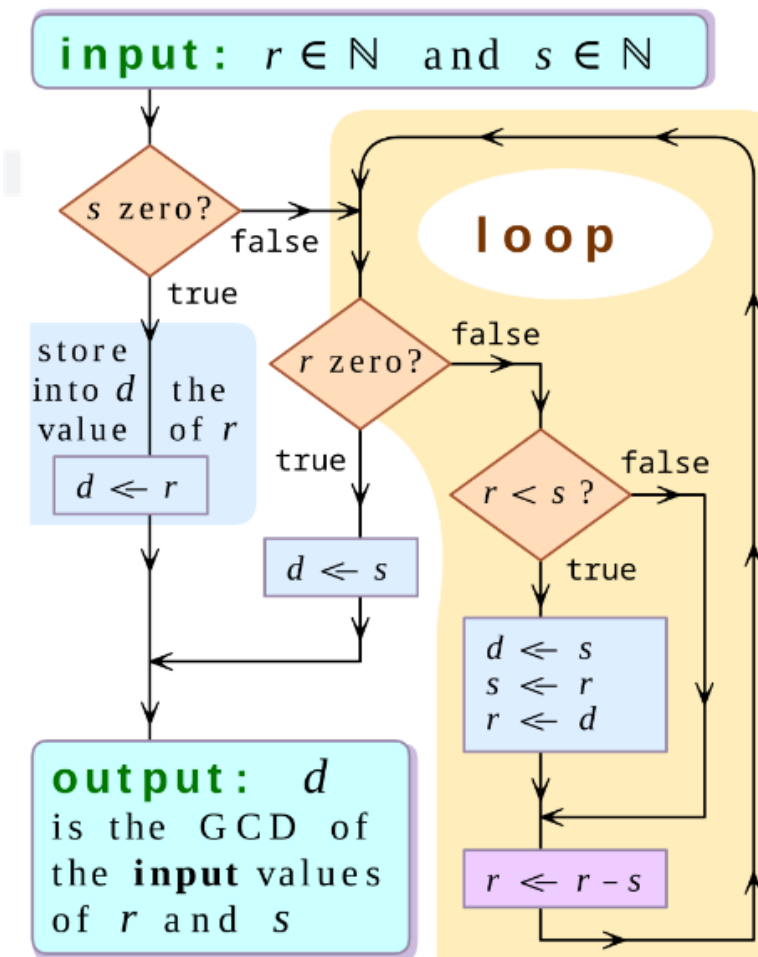


HCI

Come avere il **CONTROLLO** su di loro

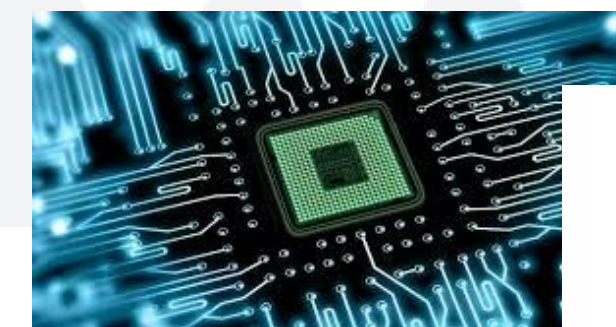


Algoritmi



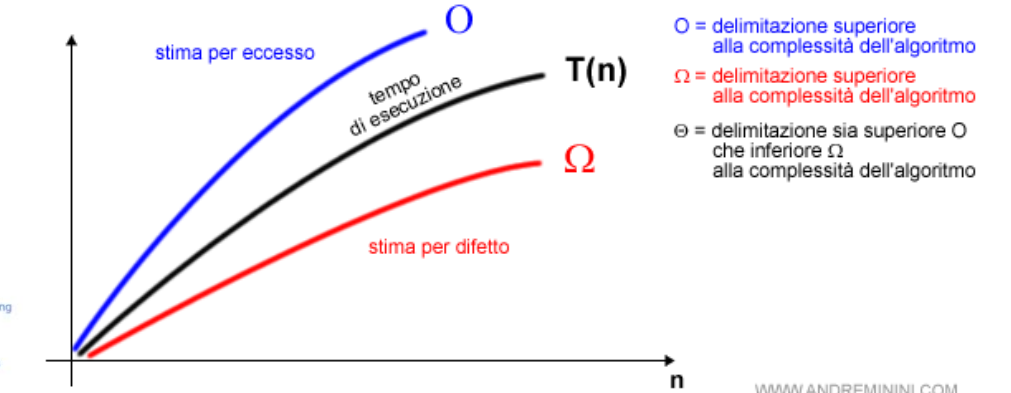
Complessità computazionale

Come renderle **EFFICIENTI**  
e il loro uso non troppo dispendioso in termini di  
energia e tempo



Architettura e logica

Software Quality



UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE

# Programma del corso e metodologie didattiche

Principale contenuto del corso

## MODULO 1: ARCHITETTURA DEL COMPUTER

- ❖ Rappresentazione dell'informazione
- ❖ Logica per l'elaborazione e componenti del computer
- ❖ Flusso dati (datapath)



## MODULO 2: STRUTTURE DATI E ALGORITMI

- ❑ Strutture dati (array, liste, pile, code, alberi, tabelle hash)
- ❑ Algoritmi: ricorsione, ordinamento, ricerca cammino minimo
- ❑ Accenni di Teoria della complessità



## INTRODUZIONE AL MACHINE LEARNING e AI

- ❑ Introduzione ai concetti di AI, deep learning, machine learning

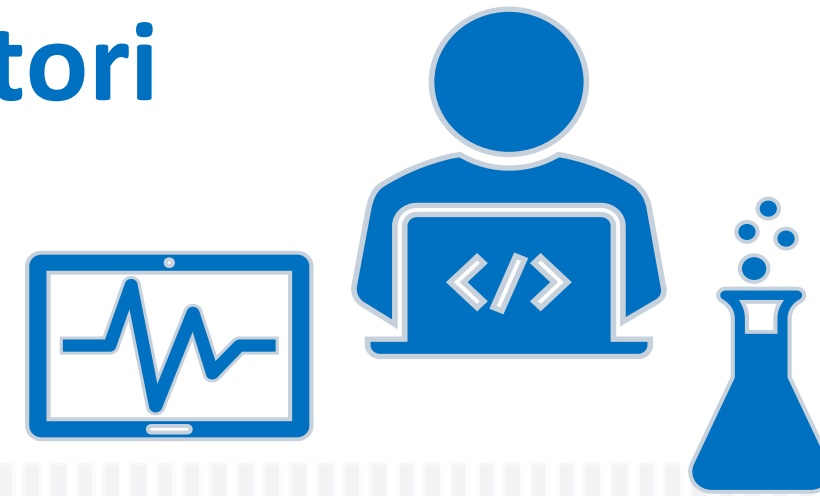


# Modalità didattiche

**Lezioni frontali**  
50h



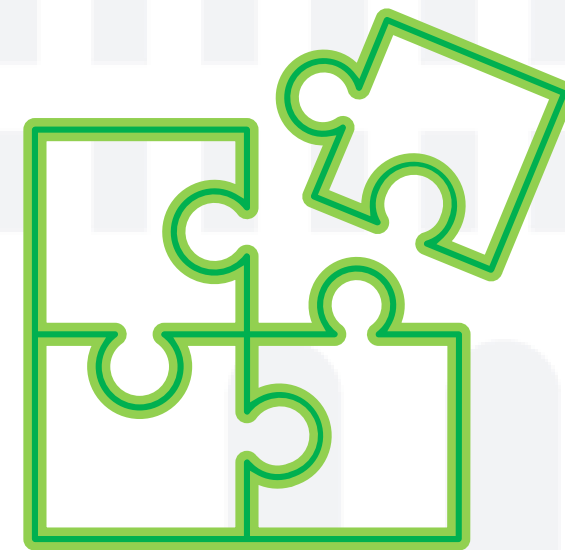
**Laboratori**  
16h



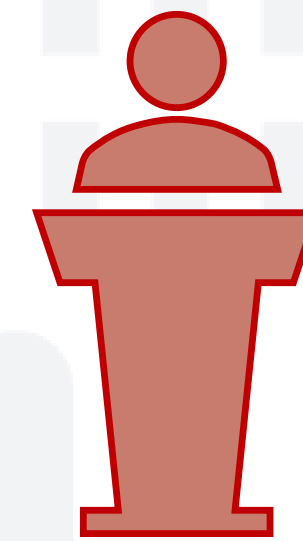
**Discussioni in classe e  
durante i lab**



**Sessioni jigsaw**  
2h



**Seminari**  
4h



# Esami (1/2)

## Due modalità



Per accedere all'orale, il voto dello scritto o della somma dei parziali deve essere sufficiente (ovvero  $\geq 18/30$ ).



PARTI	CONTRIBUTO AL VOTO FINALE
2 PARZIALI	40%
ESAME SCRITTO	40%
ESAME ORALE	60%

Per frequentanti (studenti in presenza)

PARTECIPAZIONE A SEMINARI	BONUS (da 0 a 1 punto)
---------------------------	------------------------

## Appelli

## Data ora (aula v.Moodle)

Appello I scritto	04/06/2026 14:00
Appello I orale	09/06/2026 09:30
Appello II scritto	23/06/2026 14:00
Appello II orale	26/06/2026 09:30
Appello III scritto	07/07/2026 14:00
Appello III orale	10/07/2026 09:30
Appello IV scritto	10/09/2026 14:00
Appello IV orale	17/09/2026 09:30

Ci saranno altri due appelli nella sessione di recupero (gennaio/febbraio'27).

*Stay tuned on Moodle!*



UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE

# Esami (2/2)

## Scritto (esempi di domande.. elenco non esaustivo!)

- Qualche domanda a risposta multipla
- Conversioni tra sistemi numerici
- Semplificazioni di funzioni logiche
- Conversioni tabella di verità – circuito – funzione logica
- Analisi, comprensione, completamento di un semplice programma assembly
- Analisi, comprensione, completamento di uno pseudocodice
- Pseudocodice di un algoritmo visto o di accesso/ricerca/ecc. in struttura dati
- Applicazione di un algoritmo di ordinamento/ricerca cammini minimi su grafo

## Orale

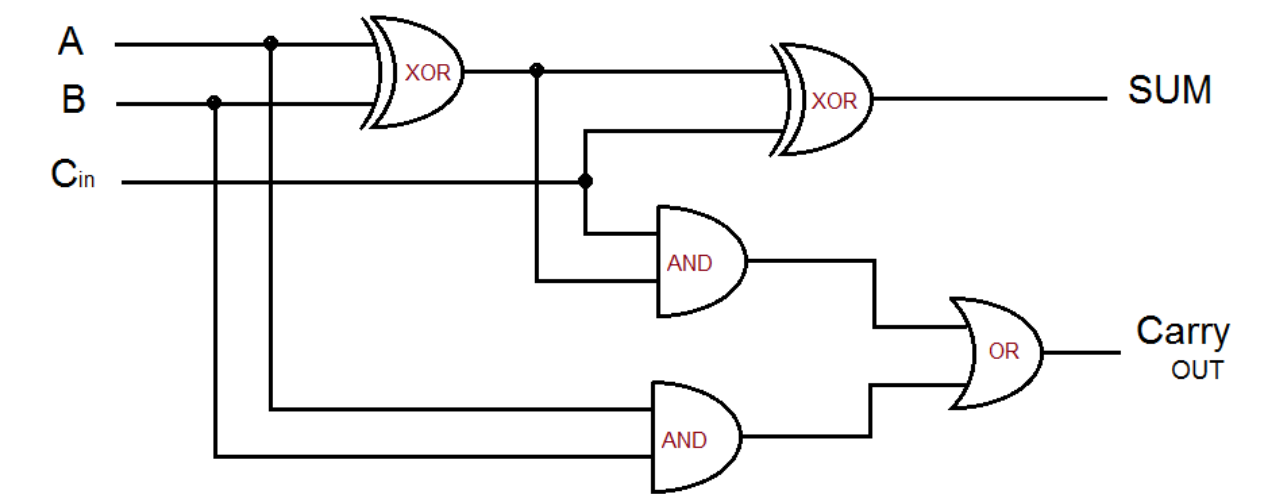
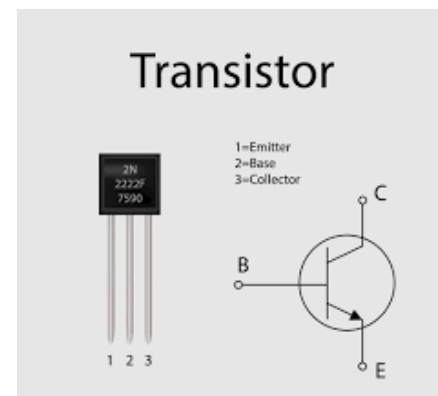
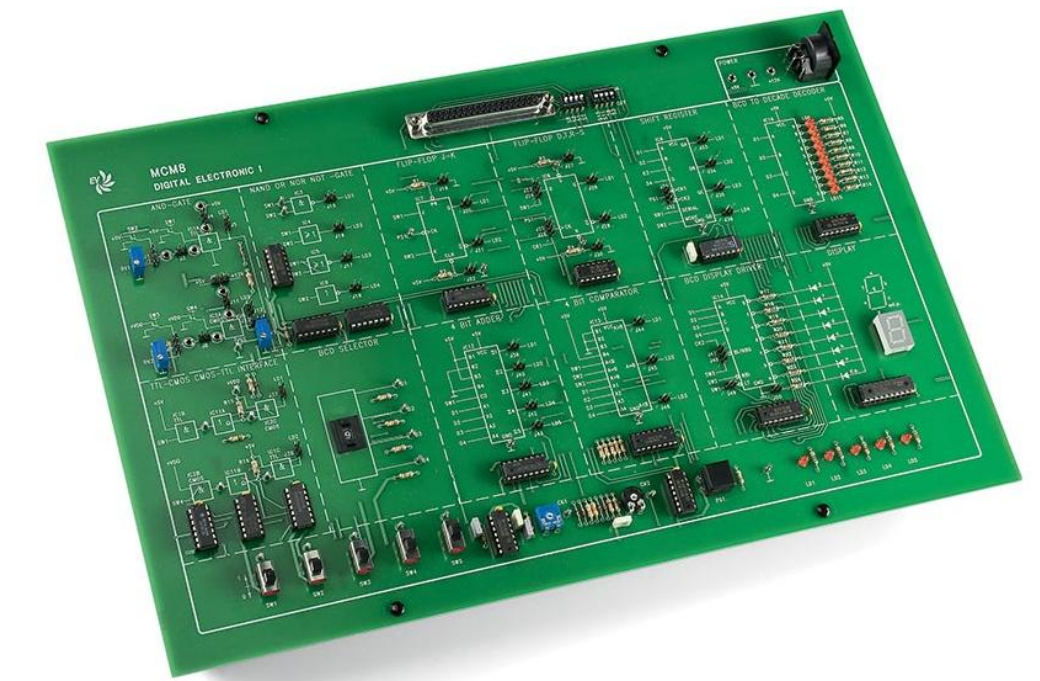
- Prima domanda da una lista di domande (non esaustiva del programma) → disponibile entro settimana prossima
- Altre domande su tutto il programma del corso

All'orale verrà valutata la **conoscenza dei contenuti del corso**, ma anche la **capacità di esprimersi in modo chiaro e non ambiguo**: esercitatevi a ripetere a voce alta o con qualcuno che vi ascolti e vi dica se capisce quello che state spiegando (*ricordatevi l'esperienza di jigsaw!*)



# Take-home messages (1/6)

- Informatica vuol dire *software ma anche hardware*
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo *più rapido e con meno errori*



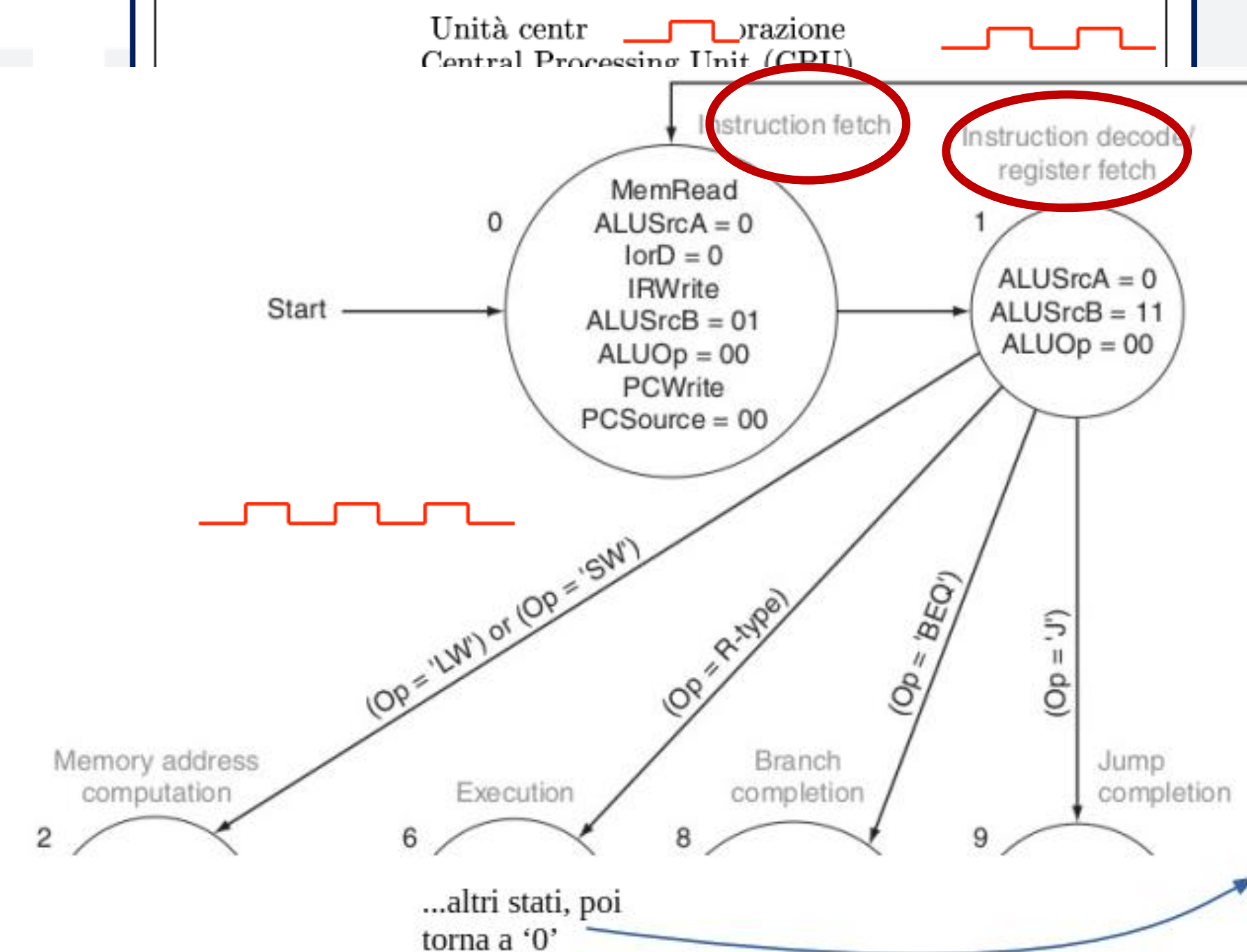
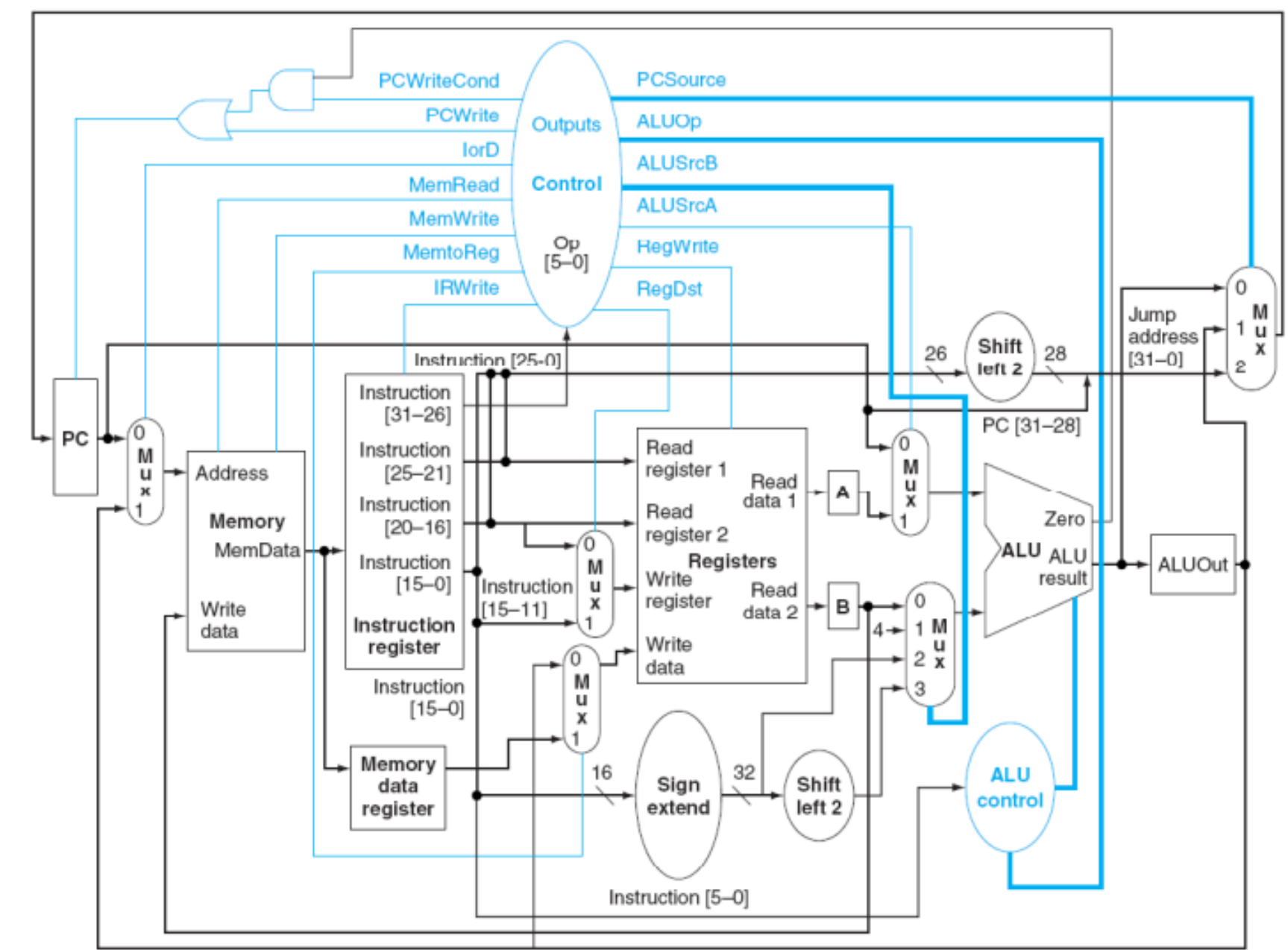
Inputs			Outputs	
A	B	C <sub>in</sub>	S	C <sub>out</sub>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = (A \oplus B) \oplus C_{in}$$



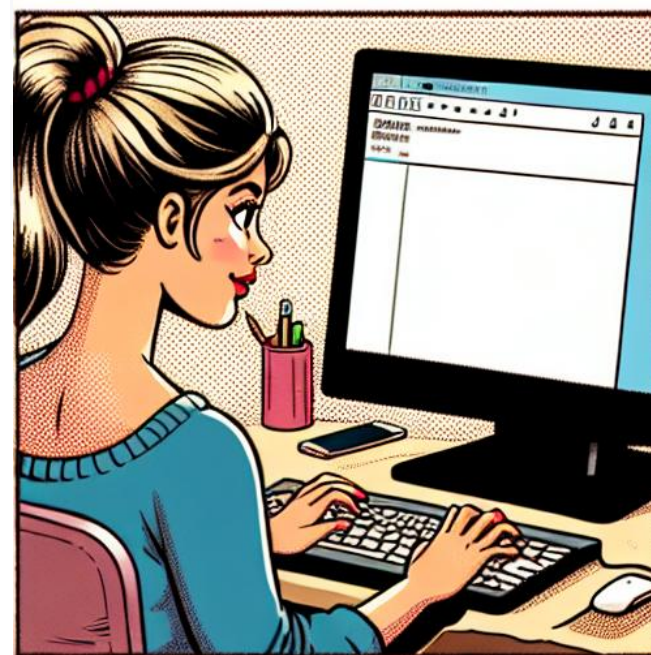
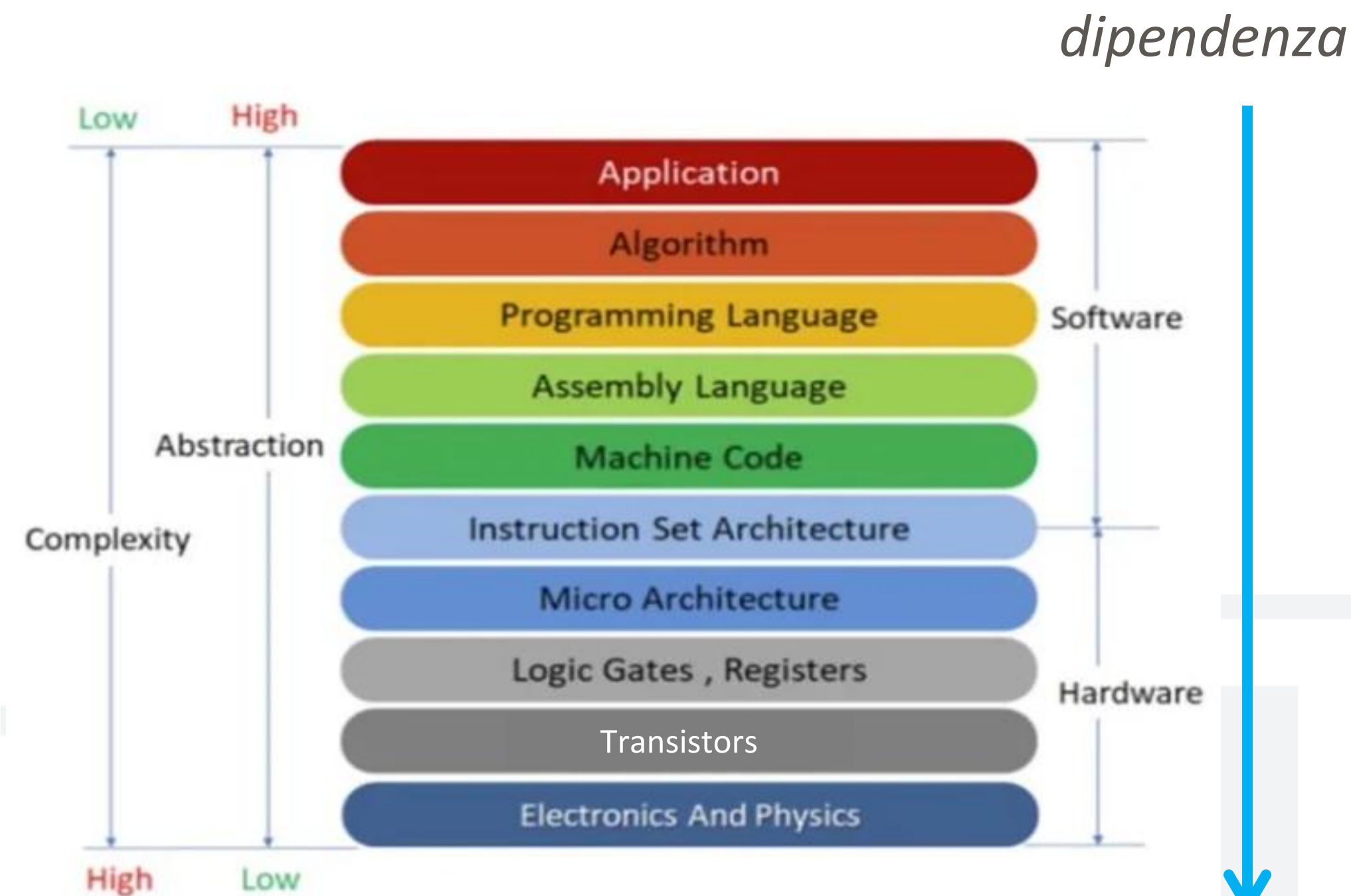
# Take-home messages (2/6)

- Informatica vuol dire *software ma anche hardware*
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo *più rapido e con meno errori*
- A seconda del tipo di hardware, si possono/non si possono eseguire determinate istruzioni (**ISA**)
- Uno stesso tipo di **architettura** può implementare molte funzioni, se progettata/programmata opportunamente



# Take-home messages (3/6)

- Informatica vuol dire *software ma anche hardware*
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo più rapido e con meno errori
- A seconda del tipo di hardware, si possono/non si possono eseguire determinate istruzioni (**ISA**)
- Uno stesso tipo di architettura può implementare molte funzioni, se progettata/programmata opportunamente
- **Esistono «linguaggi» diversi** a seconda del livello di astrazione



Linguaggio naturale (input)



Rappresentazione dell'informazione (Assembly, binario, ..)

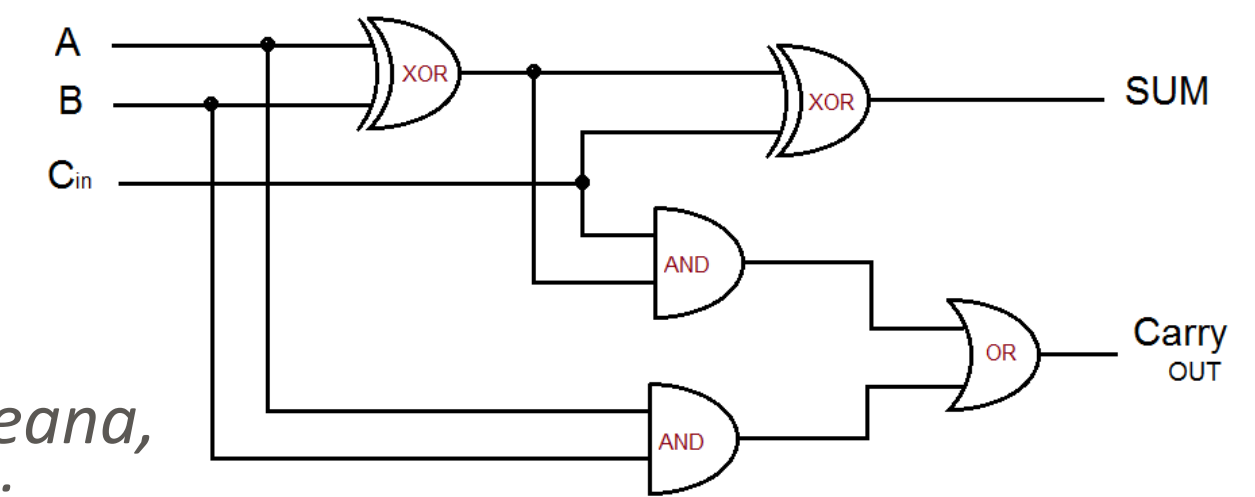


Linguaggio naturale (output)

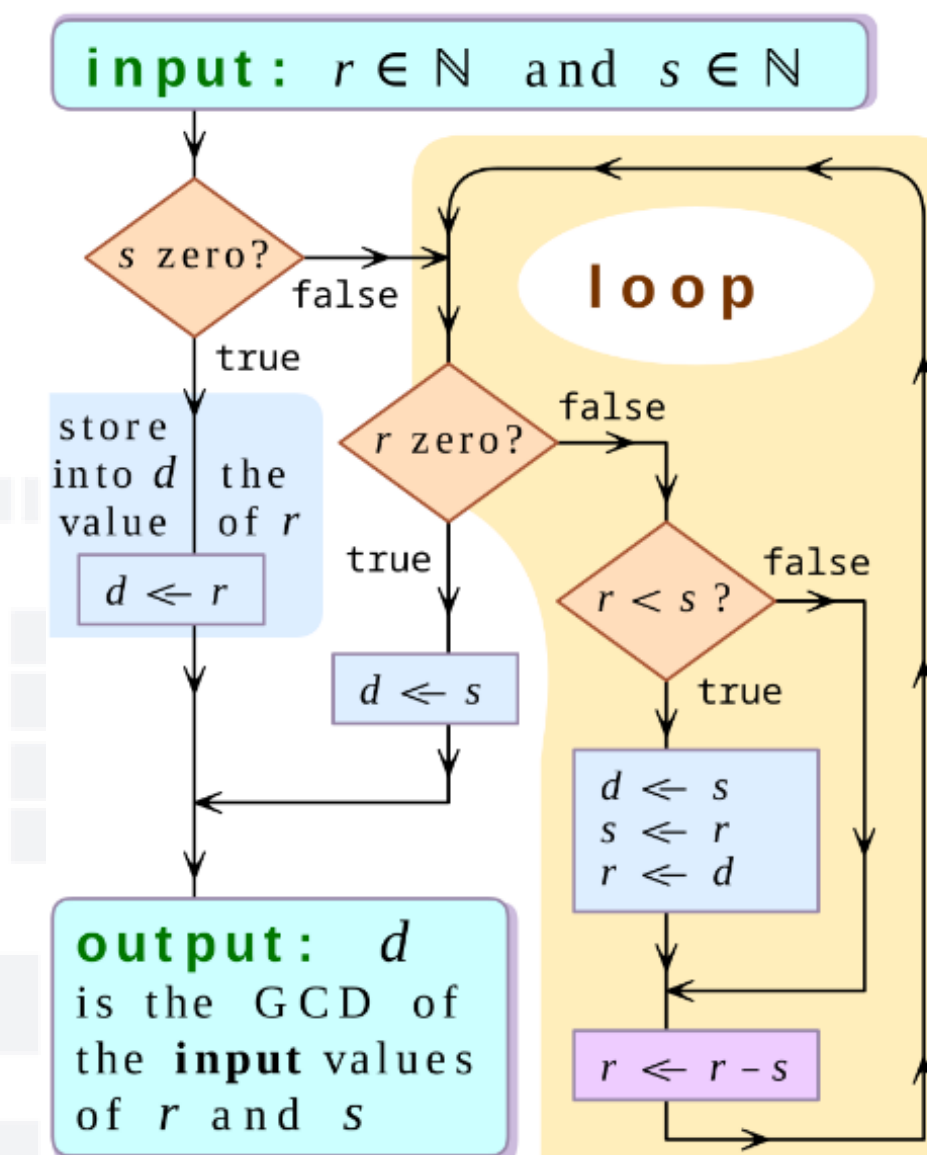
## Take-home messages (4/6)

- Informatica vuol dire **software ma anche hardware**
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo più rapido e con meno errori
- A seconda del tipo di hardware, si possono/non possono eseguire determinate istruzioni (**ISA**)
- Uno stesso tipo di architettura può implementare molte funzioni, se progettata/programmata opportunamente
- **Esistono «linguaggi» diversi** a seconda del livello di astrazione
- **Diverse rappresentazioni per lo stesso tipo di «problema» (lab#3)**
- Algoritmi «notevoli» supportati da teoremi e da studi precedenti
- Ci possono essere **più soluzioni per lo stesso problema**, ma vanno valutati vantaggi/svantaggi di una o l'altra

Algebra booleana,  
circuiti, logica,  
insiemistica

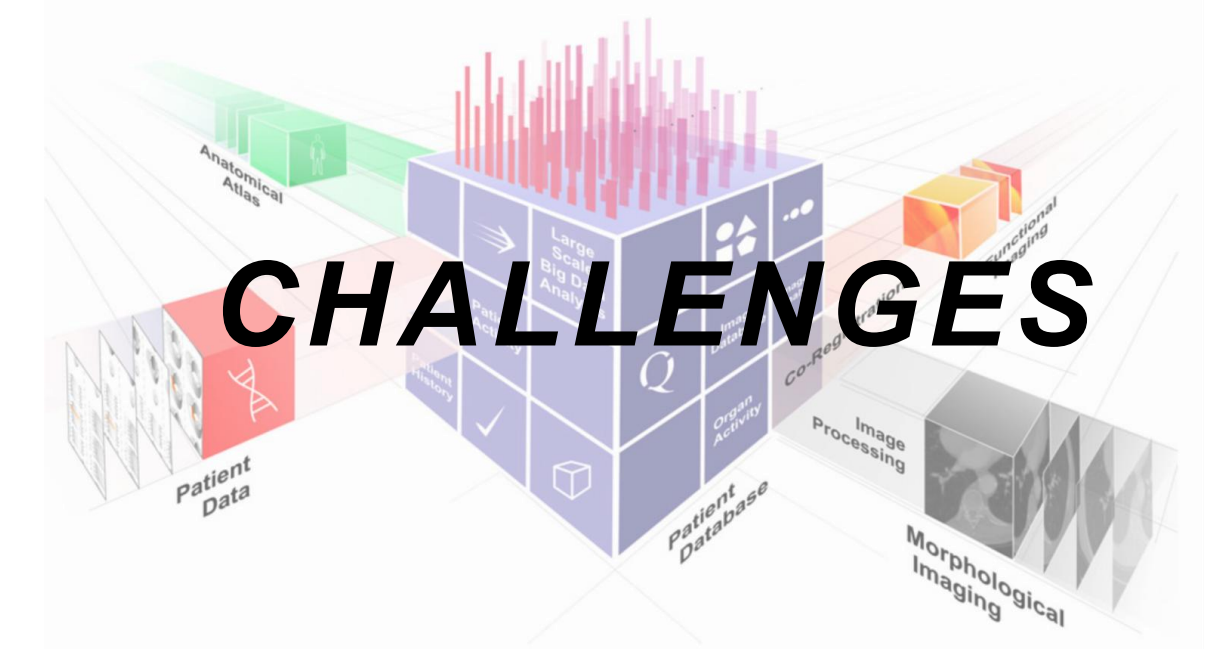


$$S = (A \oplus B) \oplus C_{in}$$

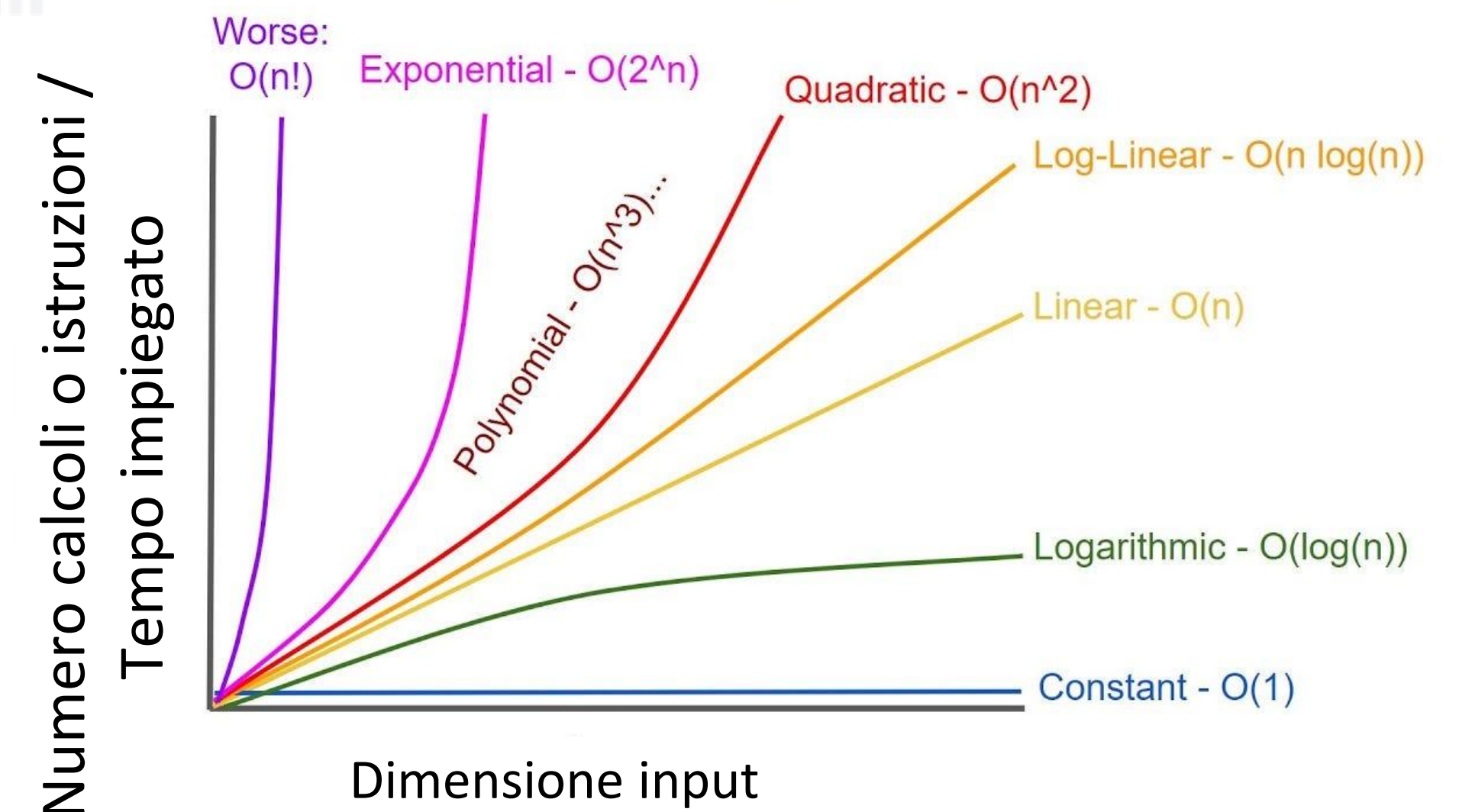


## Take-home messages (5/6)

- Informatica vuol dire *software ma anche hardware*
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo più rapido e con meno errori
- A seconda del tipo di hardware, si possono/non possono eseguire determinate istruzioni (**ISA**)
- Uno stesso tipo di architettura può implementare molte funzioni, se progettata/programmata opportunamente
- **Esistono «linguaggi» diversi** a seconda del livello di astrazione
- **Diverse rappresentazioni per lo stesso tipo di «problema» (lab#3)**
- Algoritmi «notevoli» supportati da teoremi e da studi precedenti
- Ci possono essere **più soluzioni per lo stesso problema**, ma vanno valutati vantaggi/svantaggi di una o l'altra

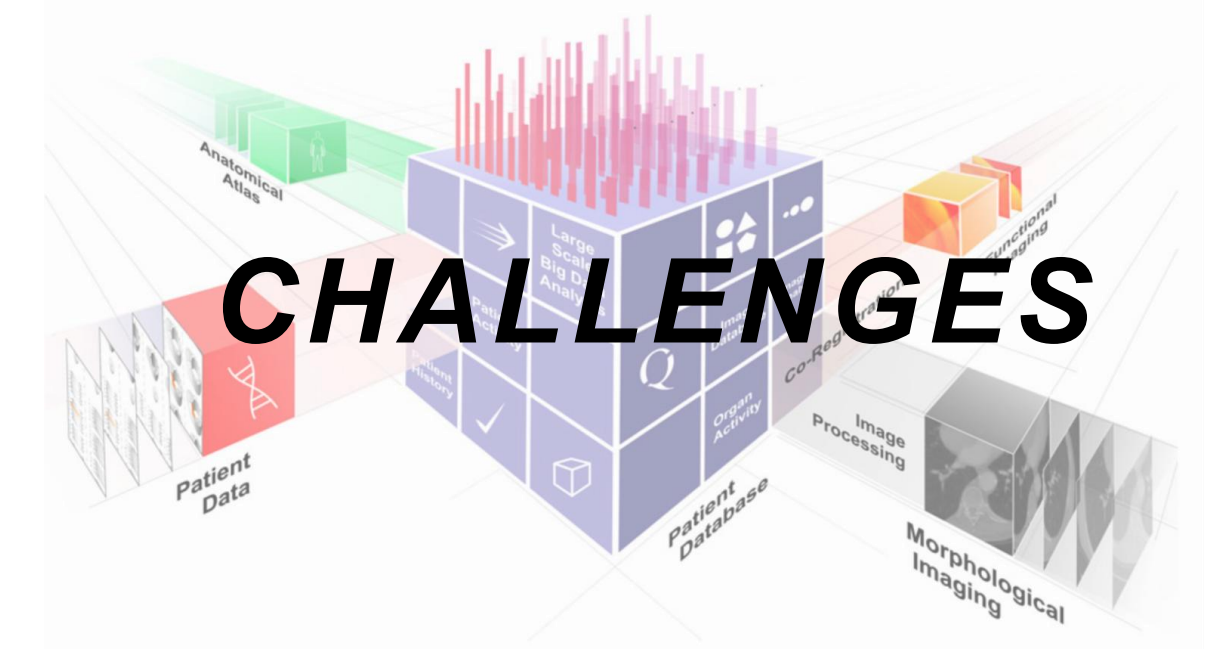


- **Risorse (tempo e spazio)**
- **Complessità**

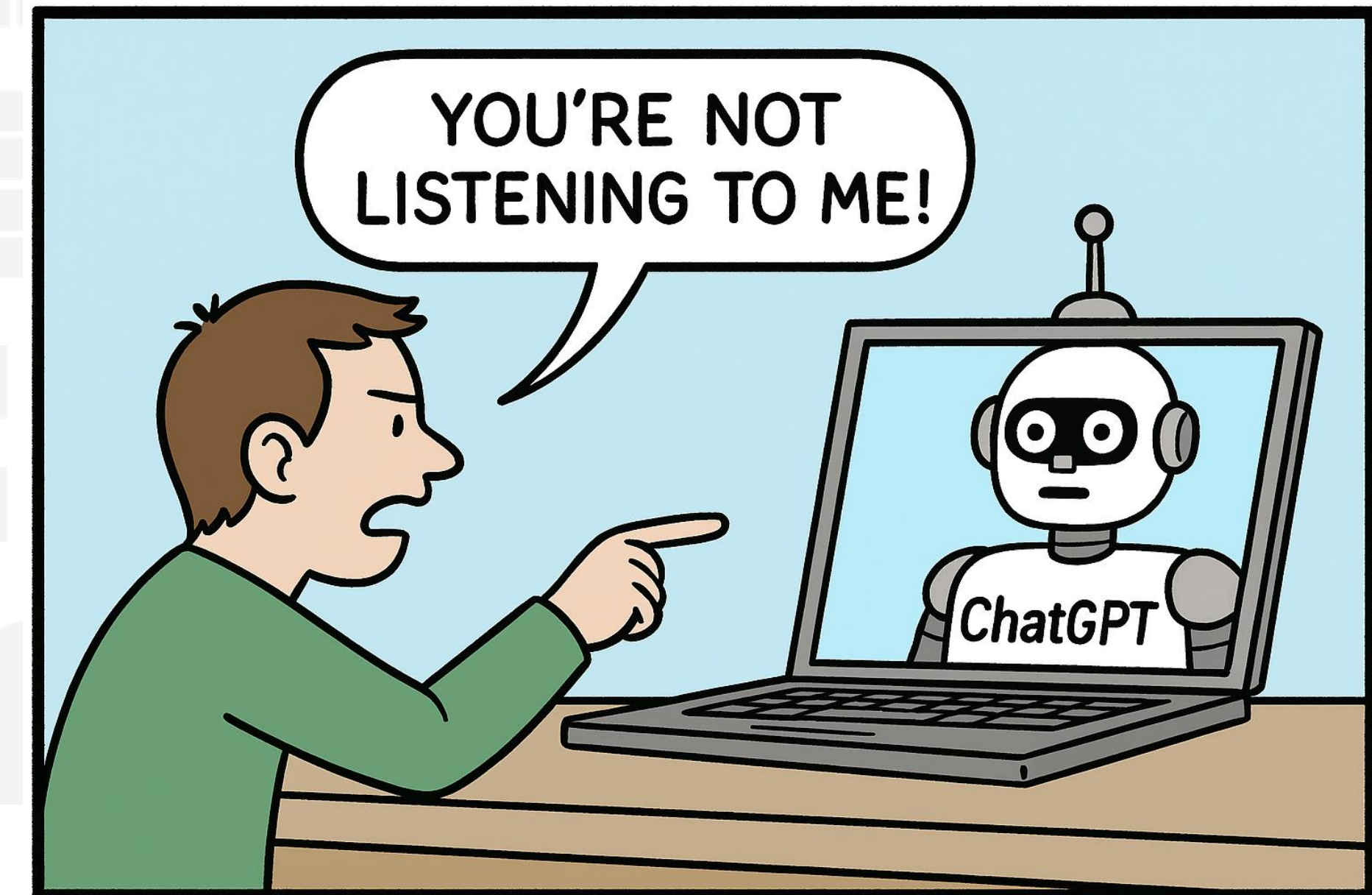


## Take-home messages (6/6)

- Informatica vuol dire *software ma anche hardware*
- Informatica vuol dire **servirsi della tecnologia per eseguire compiti ripetitivi e automatizzabili** in modo più rapido e con meno errori
- A seconda del tipo di hardware, si possono/non possono eseguire determinate istruzioni (**ISA**)
- Uno stesso tipo di architettura può implementare molte funzioni, se progettata/programmata opportunamente
- **Esistono «linguaggi» diversi** a seconda del livello di astrazione
- **Diverse rappresentazioni per lo stesso tipo di «problema» (lab#3)**
- Algoritmi «notevoli» supportati da teoremi e da studi precedenti
- Ci possono essere **più soluzioni per lo stesso problema**, ma vanno valutati vantaggi/svantaggi di una o l'altra
- **Algoritmica tradizionale vs AI**



- **Risorse (tempo e spazio)**
- **Complessità**
- **Algoritmica tradizionale vs AI**



# ESEMPI DI ESERCIZI PER II PARZIALE/SCRITTI

Argomenti del II parziale:  
Da lezione 18/03 (circuiti  
sequenziali)

11. Completare (###) e spiegare brevemente lo pseudocodice di mergesort. Poi simulare la sua esecuzione per ordinare l'array [8 5 19 20 3 12].

```
MERGE-SORT(A,p,r)
if p ≥ r
  ###
  q = ###
  ###
  ###
```

```
MERGE(A,p,q,r)
nL = q-p-1
nR = r-q

for i=0 to nL -1
  ###
for j=0 to nR -1
  ###

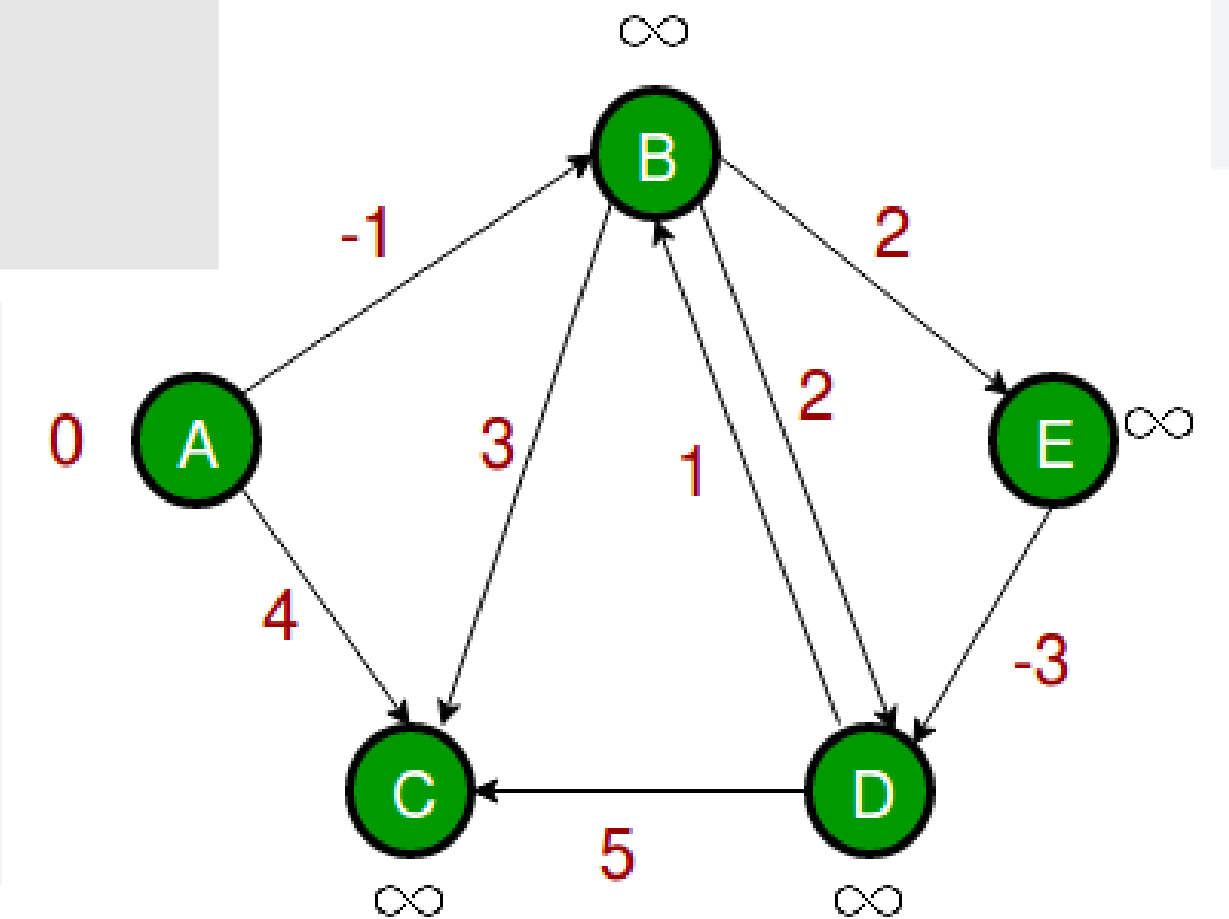
i=0 //indice che scorre su L
j=0 //indice che scorre su R
k=0 //indice che scorre su A

while i ≤ nL & j ≤ nR
  if L[i] ≤ R[j]
    A[k] = L[i]
    i = i+1
  else
    A[k] = R[j]
    j = j+1
  k = k+1
while ### //finisco di copiare il resto di L in A
while ### //finisco di copiare il resto di R in A
```

11. Interpretare il seguente pseudocodice (ovvero darne una breve, ma precisa, descrizione). Poi applicarlo sull'array A = [5 3 7 2 2 6 0 1] e determinarne la complessità computazionale.

```
// Input: array A, numero intero k
B = array di k elementi inizialmente tutti a zero
C = array di len(A) elementi
for i=0 to len(A)-1
  B[A[i]] = B[A[i]] + 1
for i=1 to k-1
  B[i] = B[i] + B[i-1]
for i=len(A)-1 downto 0
  C[B[A[i]]-1] = A[i]
  B[A[i]] = B[A[i]] - 1
return C
```

Dato il seguente grafo e dato A come nodo sorgente, trovare tramite l'algoritmo di Bellman-Ford, il cammino minimo tra il nodo sorgente e ogni altro nodo del grafo.



Dato un certo scenario, descrivere la FSM e scrivere le funzioni di aggiornamento dell'output e dello stato

Domande teoriche su datapath, FSM, ...

Testi e soluzioni saranno rese disponibili per il tutorato di martedì 20/05



UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE

# QUESTIONARIO UFFICIALE (PER REGISTRARSI AGLI ESAMI)

- *Importante, da fare con cura!*
- Tenete conto la seguente tabella di conversione:

SCALA 0-10	SCALA 0-30
0	0
1	3
2	6
3	9
4	12
5	15-17
6	18-20
7	21-23
8	24-26
9	27-28
10	29-30

Appena sufficiente →  
Mediocre o buono →  
Ottimo o eccellente →



Grazie al **Dott. Matteo Gallo** per il lavoro svolto, la competenza e la disponibilità!



Grazie a tutti per l'attenzione e la partecipazione attiva e.. *in bocca al lupo!*



**UNIVERSITÀ  
DEGLI STUDI  
DI TRIESTE**

***Seminario Prof. Ghidoni: 19 maggio, h.14:00, aula 4C***