

FONDAMENTI DI INFORMATICA

Alessandro Renda

Dipartimento di Ingegneria e Architettura, Università degli Studi di Trieste

INTRODUZIONE

Anno Accademico 2024/2025

Introduzione

- [013IN] Fondamenti di Informatica
- Insegnamento obbligatorio
- Primo anno, secondo semestre
- Durata: 72 ore (48 lezione, 24 laboratorio)
- Corso di studi: **Ingegneria Industriale (L-9)**

Introduzione

	Lunedì	Martedì
09-10	TEORIA	TEORIA
10-11		
11-12	LABORATORIO	
12-13		

- Le lezioni inizieranno alle 09.15

Introduzione



Laurea Magistrale @UniPi

Ingegneria Biomedica

PhD @UniPi

Smart Computing

Ricercatore "RTD-A" @UniPi

Sistemi di Elaborazione delle Informazioni

Laurea Triennale @UniTs

Ingegneria dell'Informazione

Ricercatore "RTT" @UniTs

Sistemi di Elaborazione delle Informazioni

Contatti

- Ricevimento: su richiesta, via mail
- Contatto mail: *coming soon*

Materiale del corso

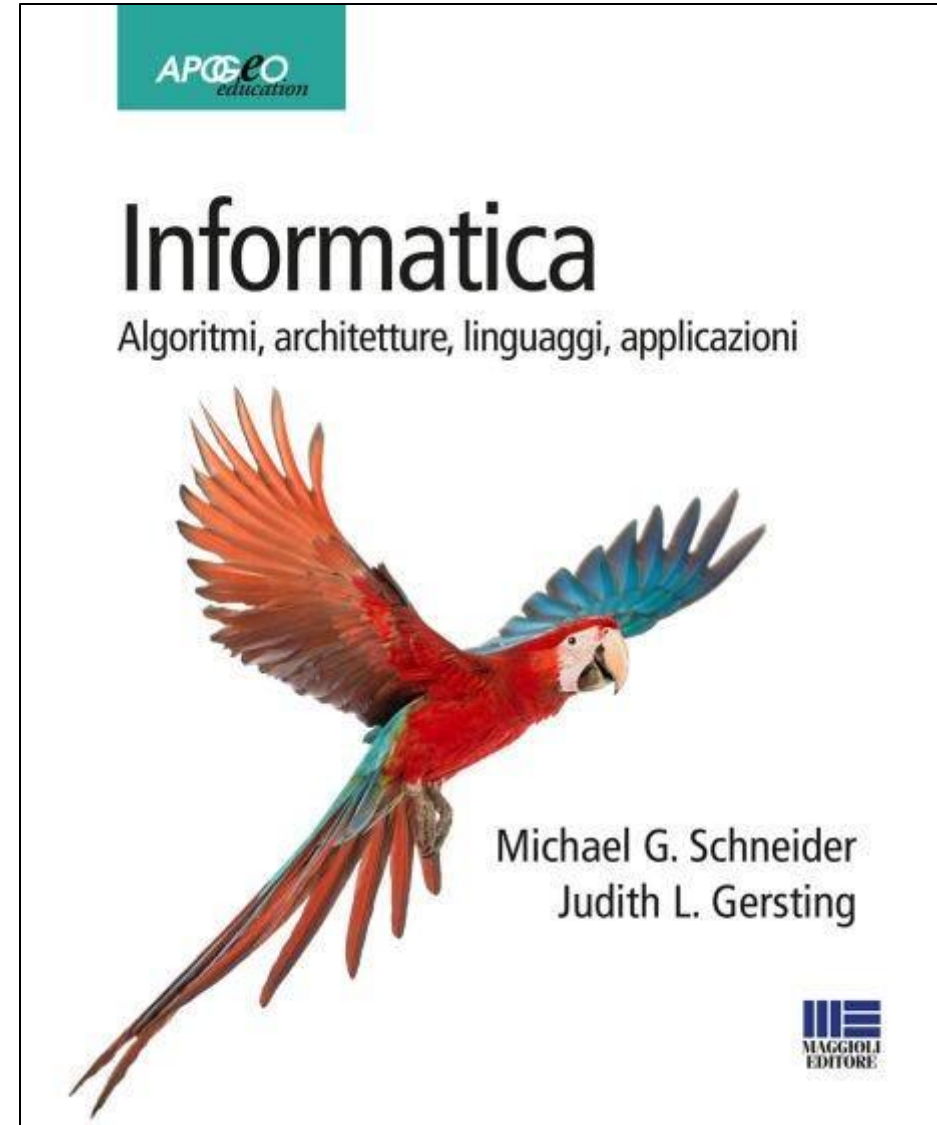
- Slide e codici del docente
- Video registrazioni delle lezioni
- Condivisione tramite piattaforma Teams *coming soon*
- *Nel frattempo*
 - il materiale verrà condiviso tramite github (auspicabilmente solo per la Settimana I)
 - <https://github.com/alerenda/013IN-UniTS>



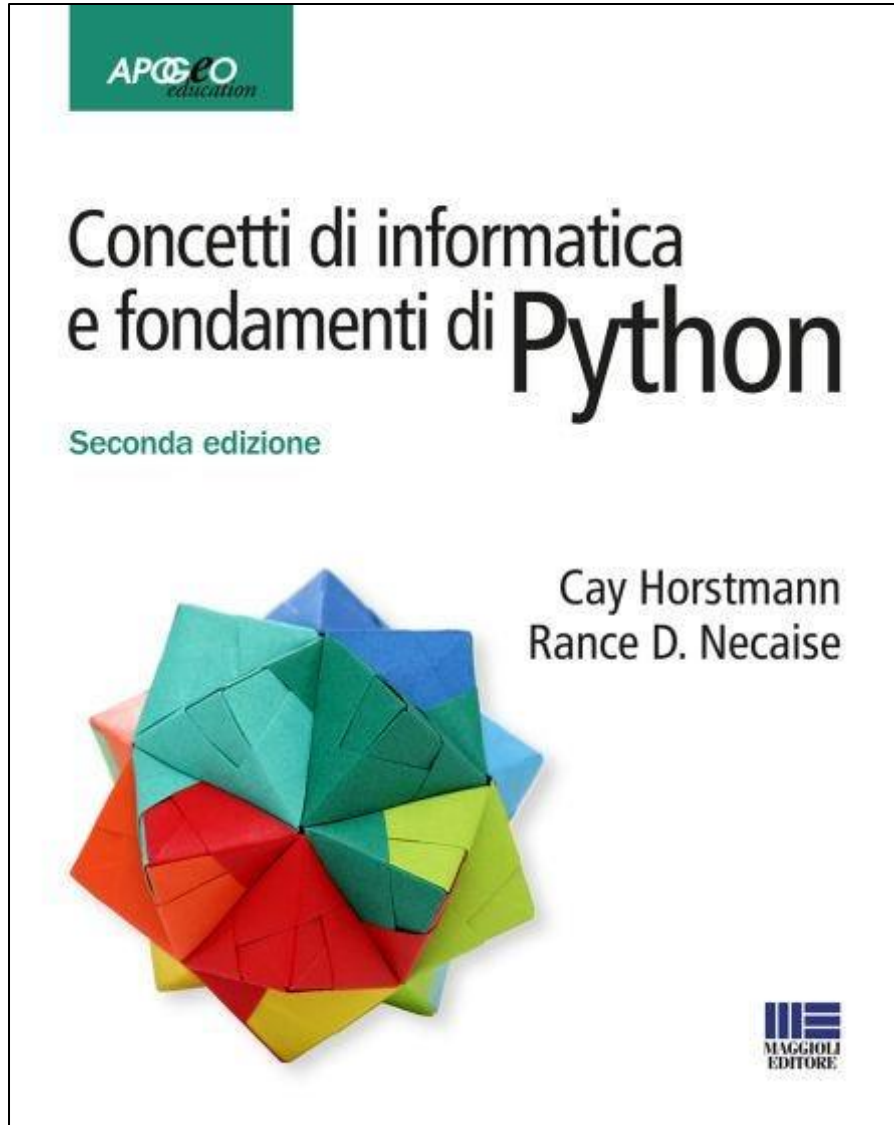
Contenuti del corso

- La rappresentazione dell'informazione
- L'hardware
- Il software
- Le reti di calcolatori
- Programmazione e linguaggio Python: **Teoria** e **Laboratorio**

Libri di Testo



Libri di Testo



Altre risorse:

- **Python in a Nutshell, 4th Edition**

- by Alex Martelli, Anna Martelli Ravenscroft, Steve Holden, Paul McGuire
- Publisher(s): O'Reilly Media, Inc.
- ISBN: 9781098113551
- 735 pagine

- **Documentazione online**

- <https://docs.python.org/3/>
- <https://docs.python.org/3/tutorial/index.html>

Ambienti di sviluppo – ne parleremo prossimamente

- Setup Locale
 - Spyder: <https://www.spyder-ide.org/>
- Tool Online
 - <https://pythontutor.com/>
 - *"It contains a step-by-step visual debugger and AI tutor to help you understand and debug code."*
 - <https://www.pythonanywhere.com/>
 - *"PythonAnywhere is a fully-fledged Python environment, ready to go, for students and teachers."*

Modalità di Esame

- Sicuramente saranno previste
 - La verifica degli aspetti *di teoria* (la rappresentazione dell'informazione, l'hardware, il software, le reti di calcolatori)
 - La verifica della capacità di risolvere problemi utilizzando il linguaggio Python
- Modalità e dettagli **in via di definizione**

Introduzione

- Una definizione di **informatica** (ACM: Association for Computing Machinery)

*Studio sistematico degli algoritmi che descrivono e trasformano l'informazione:
la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione*

Introduzione

- Una definizione (informale) di **algoritmo**

Una sequenza ben ordinata di operazioni non ambigue ed effettivamente eseguibili che producono un risultato arrivando a conclusione in una quantità di tempo finita

- Esempi
 - Istruzioni per il montaggio di un mobile
 - Ricetta per cucinare un dolce
 - Procedura per calcolare l'area di un cerchio

Introduzione

- Nota

*Se siamo in grado di specificare un algoritmo per risolvere un problema,
allora possiamo automatizzare la risoluzione del problema*

- Una definizione di **computer** (calcolatore)

*Macchina che può essere programmata
per eseguire sequenze di operazioni aritmetiche / logiche di base*

Introduzione

- Ruolo centrale del concetto di **informazione**
 - La stessa informazione può essere portata su supporti differenti



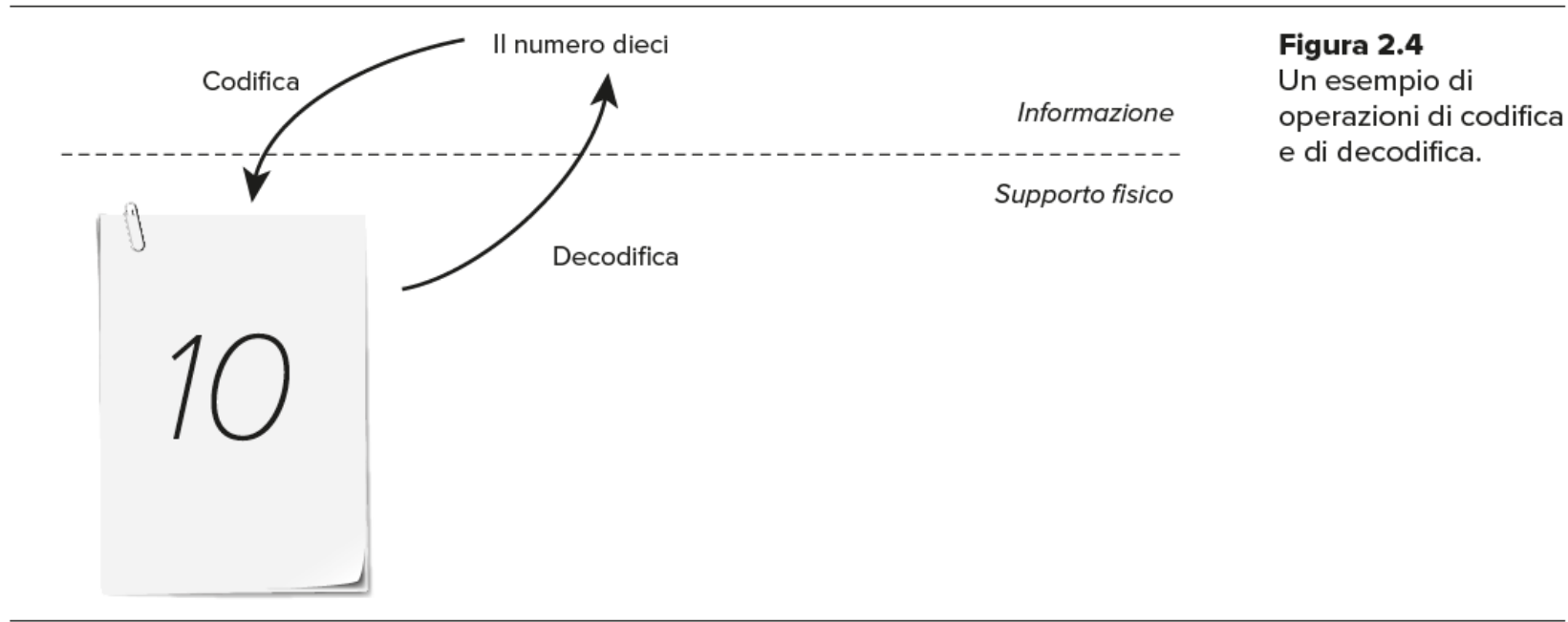
Introduzione

- Ruolo centrale del concetto di **informazione**
 - Uno stesso supporto può portare informazioni differenti



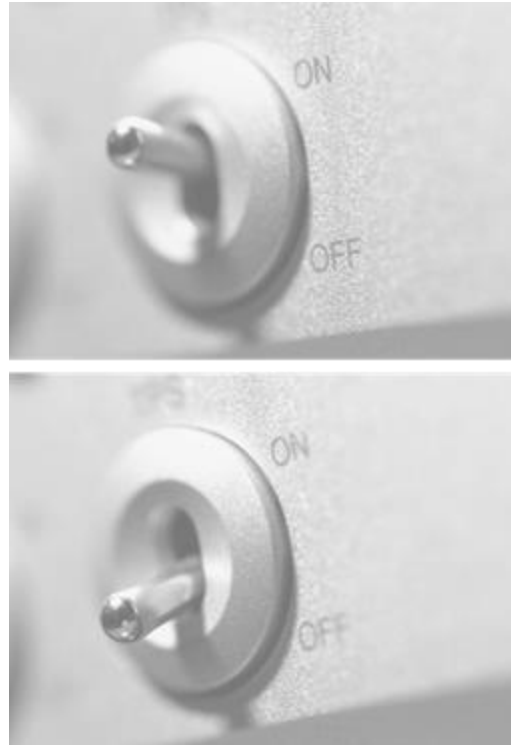
Introduzione

- **Codifica:** operazione con cui l'informazione viene scritta su supporto fisico
- **Decodifica:** operazione con cui l'informazione, su supporto fisico, viene letta



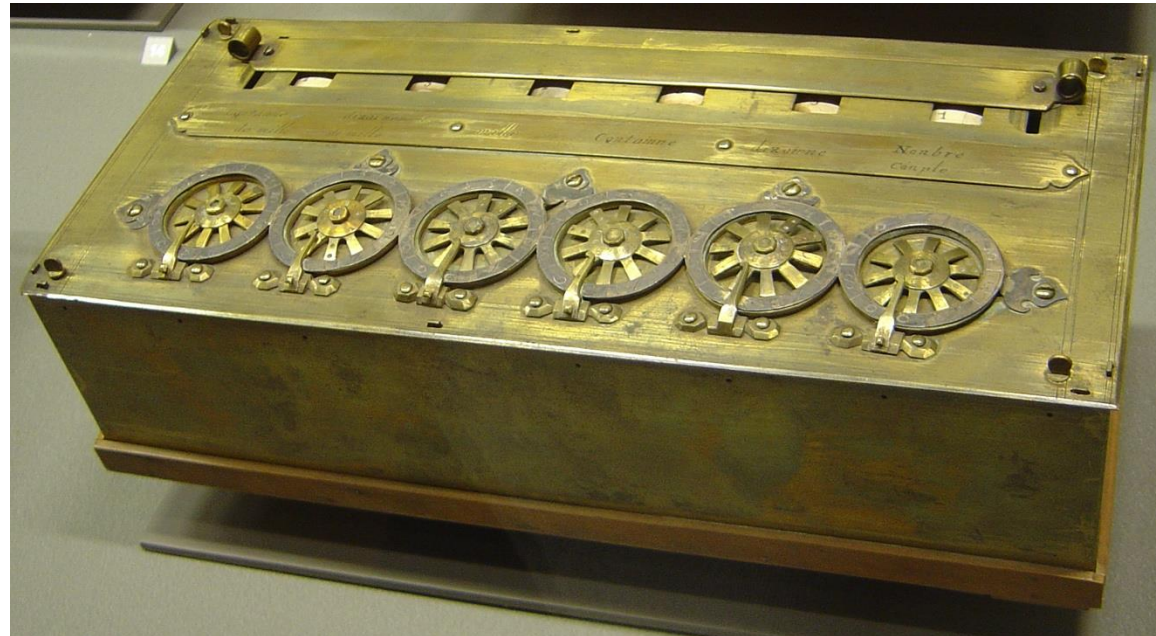
Introduzione

- Codificare i dati può essere semplice, anche usando supporti meccanici



Introduzione

- Pascal (metà del 1600):
 - una delle prime calcolatrici meccaniche
 - principalmente in legno e metallo
 - somme e addizioni



Introduzione

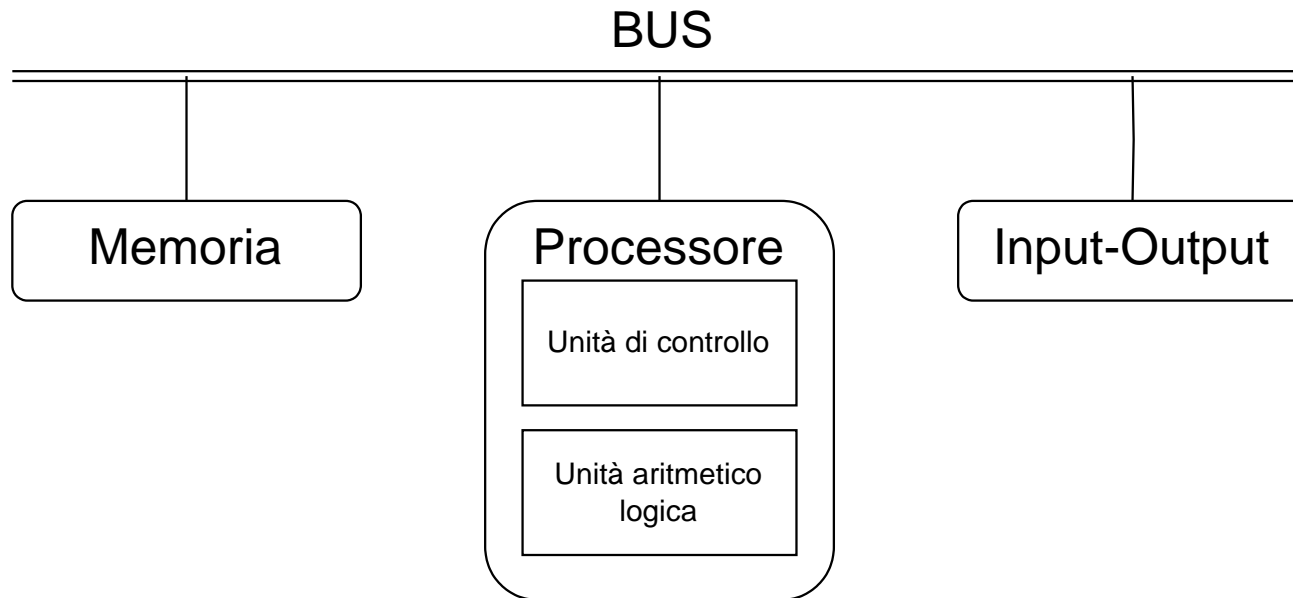
- Cosa manca alla *Pascalina* per essere un computer
 - **Memoria**
 - per archiviare le informazioni in forma leggibile dalla macchina
 - **Possibilità di essere programmata**
 - in anticipo
 - per eseguire una sequenza di istruzioni
 - senza intervento manuale

Introduzione

- Nel 1940 (300 anni più tardi) i primi calcolatori completamente elettronici
 - Ad esempio ENIAC (Electronic Numerical Integrator And Calculator)
 - Estremamente più veloci rispetto a quelli meccanici
 - Disponevano di memoria, ma essa *conteneva solo i dati e non le istruzioni*
 - Erano programmabili, ma solo *esternamente*
 - Le istruzioni venivano codificate nella struttura fisica del sistema
 - Per un problema diverso, era necessario ri-cablare l'intero calcolatore, cioè intervenire su (migliaia di) fili, connettori, quadri di connessione

Introduzione

- Modello di Von Neumann 1946
 - Alla base dei computer moderni, *general purpose*
 - *Le istruzioni che controllano il funzionamento del computer sono opportunamente codificate e memorizzate nell'unità di memoria, insieme ai dati*



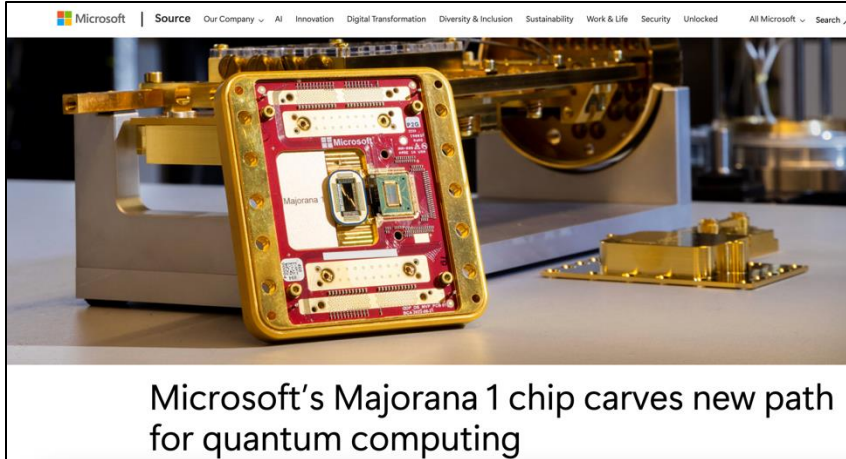
John Von Neumann

Budapest 1903 – Washington 1957

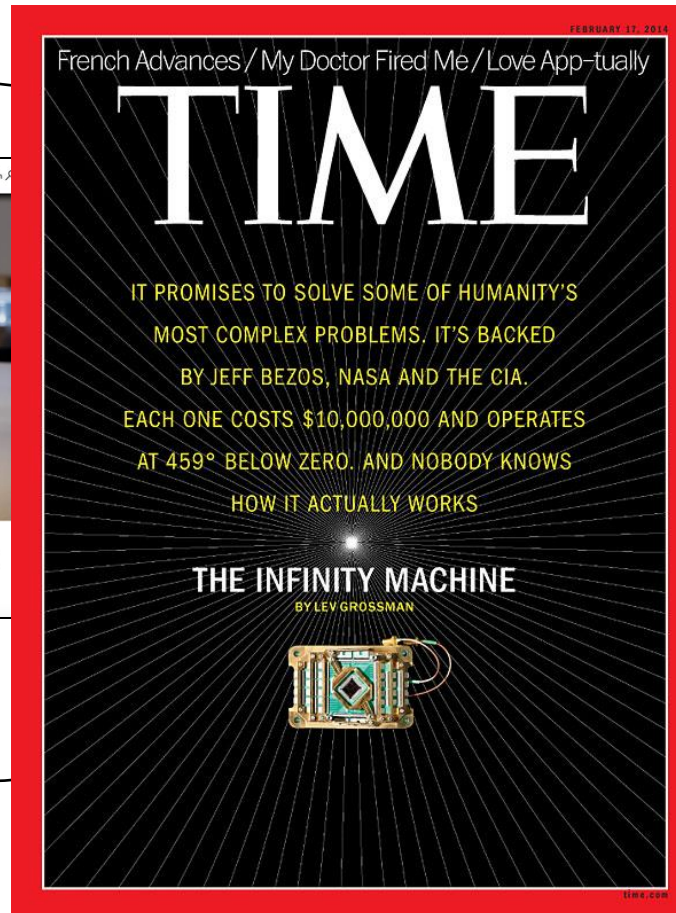


Introduzione

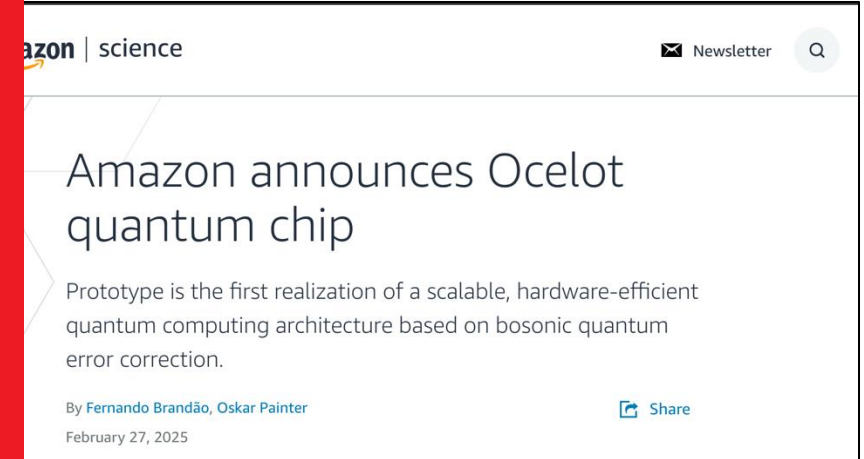
- Oltre il modello di Von Neumann ??



Pubblicato il 19-Feb-2025



Pubblicato Feb-2014



Pubblicato il 27-Feb-2025

Studenti iscritti al corso

- Corso di studi: Ingegneria Industriale (L-9)
 - Energia elettrica e dei sistemi
 - Gestionale
 - Materiali
 - Meccanica
 - Processo

Studenti iscritti al corso: un primo esempio

Quali sono le 3 squadre più tifate dagli studenti di fondamenti di informatica?

Studenti iscritti al corso: un primo esempio

Quali sono le 3 squadre più tifate dagli studenti di fondamenti di informatica?

- Di che sport? **Calcio**
- Squadre di serie A, B, C, Premier League, Nazionali ... ? **Serie A**

Studenti iscritti al corso: un primo esempio

Quali sono le 3 squadre più tifate dagli studenti di fondamenti di informatica?

- Di che sport? **Calcio**
- Squadre di serie A, B, C, Premier League, Nazionali ... ? **Serie A**

- Iscritti al corso? a che data?
Presenti in aula

Studenti iscritti al corso: un primo esempio

Quali sono le 3 squadre più tifate dagli studenti di fondamenti di informatica?

<https://forms.gle/CqJMipuUCv4gQNoc8>



Studenti iscritti al corso: un primo esempio

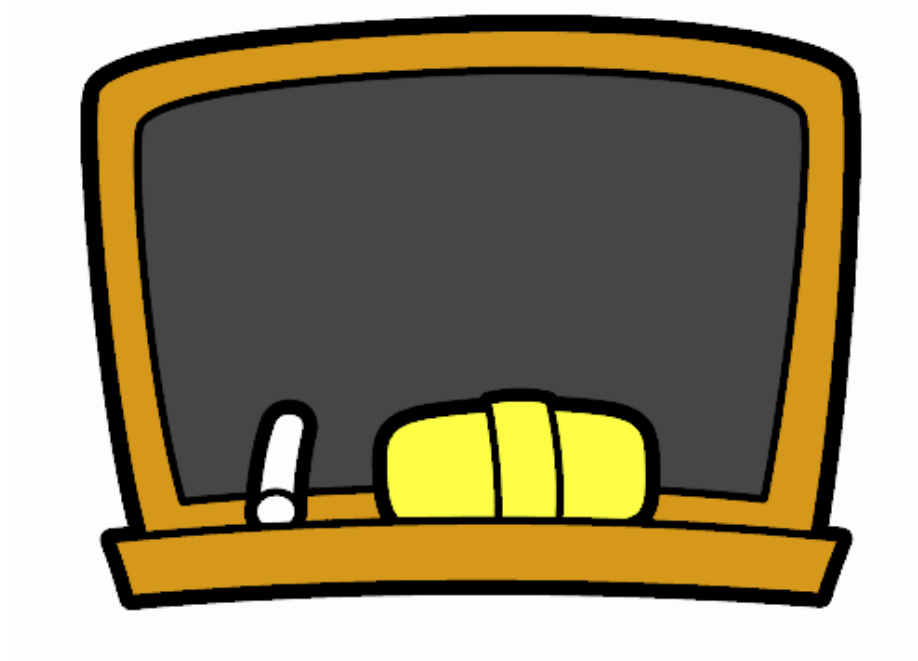
I. Analisi e formalizzazione del problema

- *Tramite il form, ogni studente in aula ha potuto indicare una squadra di calcio di Serie A della stagione 2024/2025 per cui fa il tifo, oppure selezionare l'opzione "Nessuna delle precedenti / Non seguo il calcio / Non mi interessa"*
- *Sulla base del risultato del survey, si vuole determinare quali sono le 3 squadre con il maggior numero di tifosi, e riportare per ciascuna di esse il nome della squadra ed il numero di tifosi, in ordine decrescente*

- E se tutto il tifo è diviso su 2 squadre?
- E se ci sono più squadre a condividere il terzo gradino del podio??

Studenti iscritti al corso: un primo esempio

2. Progettazione della soluzione



Studenti iscritti al corso: un primo esempio

2. Progettazione della soluzione

conteggio_squadre = vuoto

Per ogni riga del file

 squadra_corrente = leggi il nome della squadra

 Se squadra_corrente != "Nessuna delle precedenti / Non seguo il calcio / Non mi interessa»

 se squadra_corrente in conteggio_squadre

 incrementa frequenza tifo di 1 per squadra_corrente

 altrimenti

 aggiungi squadra_corrente a conteggio_squadre con frequenza tifo = 1

squadre_ordinate = ordina conteggio_squadre in base alla frequenza tifo di ciascuna, in ordine decrescente

ranking = 1

Per ogni squadra e frequenza tifo in squadre_ordinate

 stampa ranking, squadra, frequenza tifo

 ranking = ranking + 1

 se ranking > 3

 termina

Studenti iscritti al corso: un primo esempio

3. Implementazione e test

```
import csv

TIFOSI_03_03_2025 = "Prova.csv"
OPZIONE_NULLA = "Nessuna delle precedenti / Non seguo il calcio / Non mi interessa"
N = 3

def main():
    risposte = leggi_file(TIFOSI_03_03_2025)
    squadre_tifate = [x['Che squadra tifi?'] for x in risposte if
                      x['Che squadra tifi?'] != OPZIONE_NULLA]
    squadre_freq = {}
    for squadra in squadre_tifate:
        squadre_freq[squadra] = squadre_freq.get(squadra, 0) + 1
    squadre_freq = sorted(squadre_freq.items(), key=lambda item: item[1],
                          reverse=True)
    print(f"Ecco le {N} squadre più tifate:")
    for enu, (squadra, freq) in enumerate(squadre_freq[:N]):
        print(f'{enu+1:2}. {squadra:15} {freq:>4}')

def leggi_file(percorso_file):
    with open(percorso_file, 'r', encoding="utf-8", newline="") as mf:
        contenuto = list(csv.DictReader(mf))
    return contenuto

main()
```

Ecco le 3 squadre più tifate:

1. Prova	7
2. Squadra3	5
3. aaaaa	5

Studenti iscritti al corso: un primo esempio

4. Esecuzione su dati reali

Ecco le 3 squadre più tifate:

- | | |
|----------|---|
| 1. ????? | ? |
| 2. ????? | ? |
| 3. ????? | ? |

