

CONFERENZA GENERALE

DIGITALE

17/11/2025



Dott. Michael Lodi

Alma Mater Studiorum – Università di Bologna

Informatica nelle Indicazioni Nazionali 2025:

***Le grandi idee come fari
per la didattica a scuola***

www.linceiscuola.it

CONFERENZA GENERALE

Informatica e *Cultura Digitale*

17/11/2025



Dott. Michael Lodi

Alma Mater Studiorum – Università di Bologna

Informatica nelle Indicazioni Nazionali 2025:

***Le grandi idee come fari
per la didattica a scuola***

www.linceiscuola.it

Michael :-)



Ricercatore (a tempo *molto* determinato) in (Didattica dell') Informatica

Faccio ricerca sulla Didattica dell'Informatica

- Epistemologia dell'Informatica (e del *pensiero computazionale*)
- Introduzione alla programmazione
- Didattica di argomenti avanzati come la crittografia, anche interdisciplinare
 - anche Didattica di High Performance e Quantum computing

Insegno Programmazione in Python, Didattica dell'Informatica a LM e 60cfu

Sono abilitato A-41 (Scienze e tecnologie Informatiche) e vincitore rinunciatario di concorso

Ho iniziato a parlare di Informatica sui social, con il nome «Pensare informatico» (lodi.ml/social)

Tutti i riferimenti che citerò



<https://lodi.ml>

Il contesto



Le Nuove Indicazioni Nazionali
(versione 7 luglio 2025 –
parere favorevole consiglio di Stato appena arrivato)

per il primo ciclo (primaria e secondaria di primo grado)

includono competenze, obiettivi e conoscenze di Informatica
(nelle discipline Matematica e Tecnologia)

Di cosa *non* parleremo oggi, ma...

- Analisi degli aspetti principali di traguardi e obiettivi di Informatica, e strategie didattiche per il loro insegnamento/apprendimento
 - Ma **domani**: <https://programmailfuturo.it/notizie/webinar>



**Programma
il Futuro**

Algoritmi, dati e programmi:
l'informatica nelle nuove
Indicazioni Nazionali per la
scuola del primo ciclo

Violetta Lonati
Università di Milano

martedì 18 novembre 2025
18:00 → 19:00

#WebinarPiF



Un viaggio... da zero!

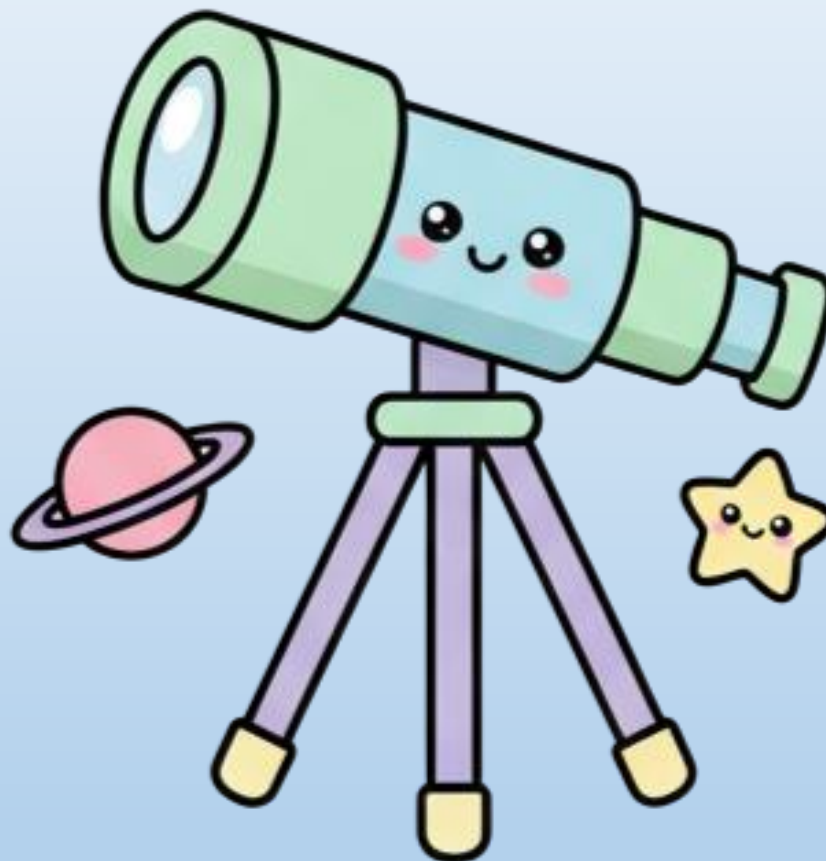


- Un viaggio (*idealmente senza prerequisiti*) alla scoperta delle idee fondamentali per poter leggere e capire meglio competenze, obiettivi e conoscenze di Informatica nelle Nuove Indicazioni
- Una visione più ampia sulle *big ideas* dell'Informatica per tutta la scuola dell'obbligo
- Una riflessione pedagogica sulle *big ideas* e sul loro ruolo nell'insegnamento

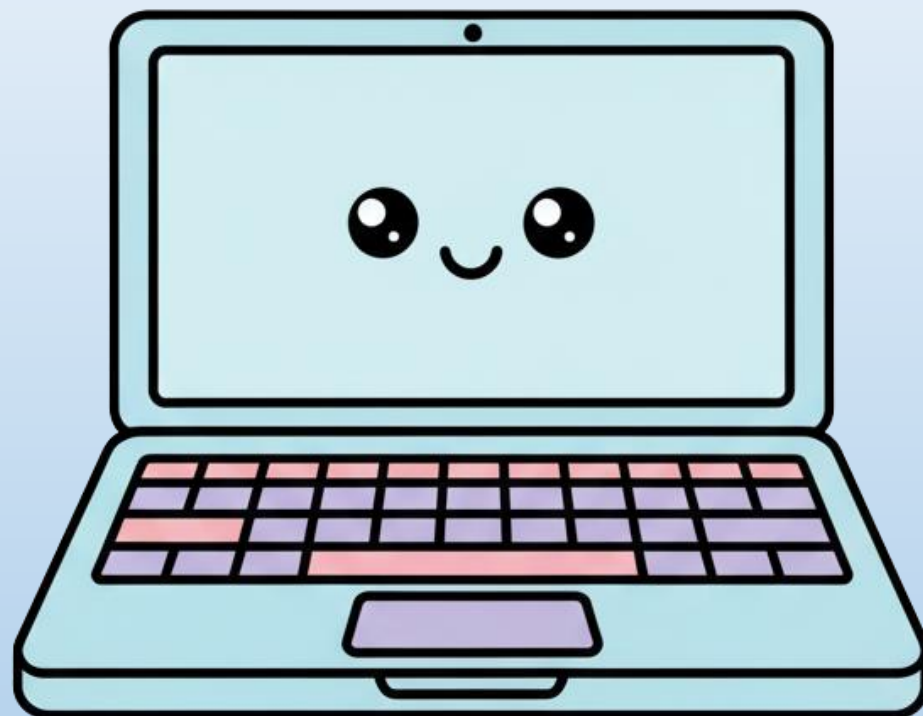
Che cos'è la biologia?



Che cos'è l'astronomia?



Che cos'è l'informatica?



Informatica

- Scienza che studia l'elaborazione automatica dell'informazione
- Informazione: conoscenza utile
- Per esprimere una informazione, posso usare dati diversi
- Posso aver bisogno di elaborare molti dati



Elaborazione automatica



- Computer (elaboratori)
- Macchine (oggi elettroniche) con interruttori e ingranaggi
- *Stato*: disposizione interna di interruttori (acceso spento)
- *Elaborazione*: accendere e spegnere gli interruttori
- *Oggi*: decine/centinaia miliardi di interruttori (transistor)
 - in numero comunque finito ($<$ atomi universo), ma «grande a piacere»

Rappresentare numeri con interruttori

$$2025 \rightarrow 2 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 5 \cdot 10^0$$

$$2025 \rightarrow 11111101001 \rightarrow 1 \cdot 2^{10} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Binary digit (*bit*)

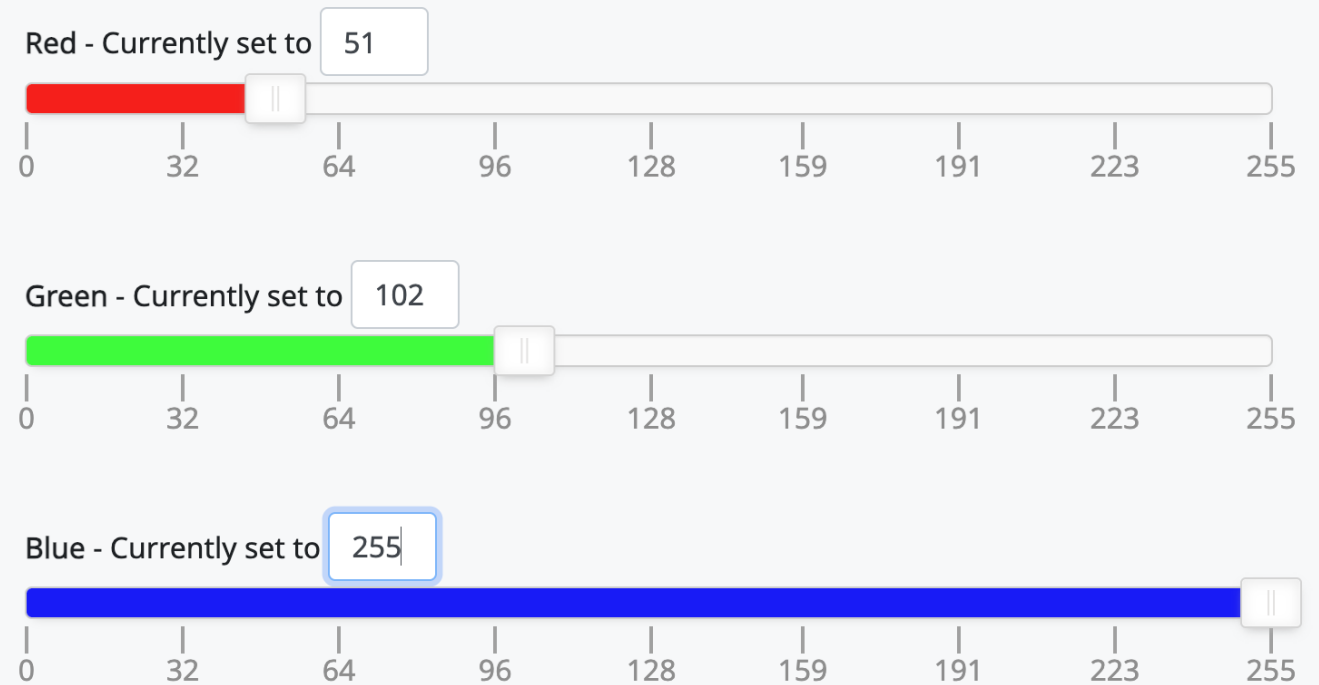
Es. 0 = spento (o bassa tensione), 1 = acceso (o alta tensione)

Rappresentare lettere e simboli

!	33	100001
A	65	1000001
a	97	1100001
b	98	1100010
Ω	937	1110101001
爱	29233	111001000110001
😊	128522	11111011000001010
...	...	...

Rappresentare colori

`rgb(51, 102, 255)`



Rappresentare immagini



Rappresentare immagini

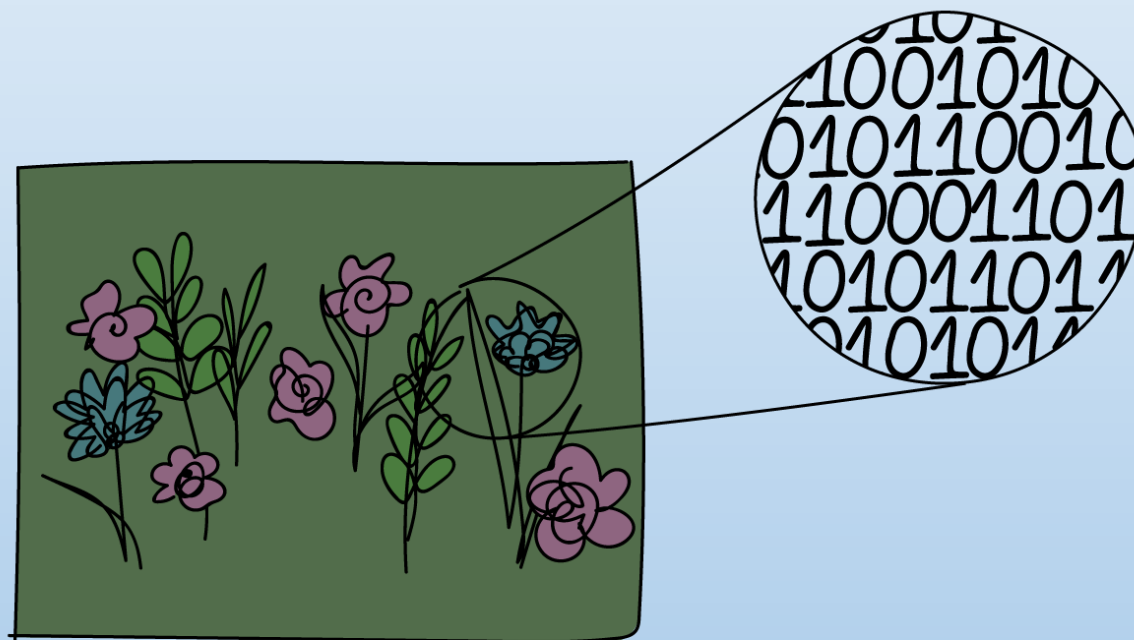


Rappresentare immagini

208	R	204	R	76	R	36	R	27	R	26	R	42	R	45	R	71	R	66	R	53	R	49	R	19	R	38	R	212	R	215	R	179	R	25	R	22
223	G	222	G	92	G	42	G	32	G	32	G	49	G	56	G	87	G	76	G	65	G	58	G	27	G	50	G	228	G	229	G	195	G	28	G	27
230	B	234	B	112	B	52	B	43	B	43	B	56	B	66	B	94	B	77	B	66	B	56	B	31	B	58	B	235	B	237	B	209	B	40	B	37
223	R	192	R	56	R	34	R	39	R	27	R	42	R	168	R	144	R	89	R	71	R	73	R	50	R	36	R	153	R	214	R	192	R	21	R	29
237	G	207	G	75	G	44	G	48	G	35	G	49	G	190	G	166	G	104	G	83	G	94	G	67	G	44	G	166	G	230	G	211	G	20	G	33
245	B	219	B	97	B	55	B	57	B	45	B	51	B	187	B	167	B	99	B	76	B	68	B	61	B	43	B	172	B	239	B	226	B	35	B	43
232	R	189	R	156	R	24	R	31	R	31	R	36	R	142	R	142	R	80	R	84	R	79	R	66	R	24	R	54	R	213	R	195	R	19	R	27
244	G	203	G	177	G	34	G	40	G	40	G	48	G	159	G	164	G	101	G	102	G	98	G	86	G	27	G	60	G	226	G	214	G	25	G	31
254	B	201	B	192	B	45	B	49	B	49	B	49	B	160	B	160	B	87	B	80	B	68	B	62	B	33	B	71	B	232	B	232	B	35	B	41
224	R	203	R	200	R	28	R	36	R	29	R	43	R	181	R	115	R	89	R	89	R	78	R	68	R	28	R	34	R	163	R	192	R	22	R	26
239	G	217	G	215	G	37	G	45	G	38	G	52	G	203	G	139	G	104	G	107	G	96	G	85	G	37	G	38	G	179	G	217	G	25	G	31
246	B	226	B	231	B	48	B	54	B	48	B	47	B	192	B	134	B	86	B	81	B	70	B	62	B	41	B	52	B	187	B	229	B	40	B	39
222	R	217	R	194	R	14	R	45	R	39	R	46	R	164	R	136	R	91	R	80	R	91	R	69	R	46	R	43	R	38	R	209	R	17	R	22
232	G	231	G	208	G	29	G	51	G	46	G	53	G	186	G	157	G	110	G	99	G	108	G	88	G	50	G	46	G	50	G	229	G	21	G	27
235	B	240	B	212	B	36	B	61	B	54	B	60	B	179	B	147	B	88	B	72	B	79	B	63	B	53	B	55	B	56	B	238	B	36	B	35
215	R	218	R	177	R	22	R	23	R	38	R	33	R	126	R	109	R	87	R	69	R	80	R	69	R	26	R	32	R	33	R	127	R	28	R	32
227	G	234	G	192	G	41	G	29	G	46	G	37	G	145	G	130	G	102	G	86	G	94	G	88	G	36	G	39	G	40	G	141	G	33	G	36
234	B	242	B	199	B	49	B	39	B	56	B	46	B	132	B	117	B	77	B	60	B	74	B	66	B	40	B	50	B	51	B	151	B	46	B	48
63	R	212	R	195	R	75	R	35	R	33	R	43	R	102	R	111	R	85	R	64	R	74	R	74	R	45	R	37	R	29	R	50	R	26	R	25
72	G	226	G	199	G	89	G	42	G	44	G	46	G	120	G	132	G	99	G	77	G	87	G	92	G	55	G	47	G	36	G	60	G	32	G	29
84	B	229	B	207	B	100	B	53	B	54	B	53	B	103	B	117	B	78	B	58	B	68	B	70	B	49	B	57	B	49	B	69	B	43	B	41
38	R	118	R	192	R	89	R	14	R	29	R	30	R	78	R	103	R	92	R	71	R	75	R	92	R	90	R	25	R	33	R	25	R	30	R	22
47	G	132	G	203	G	111	G	23	G	37	G	35	G	92	G	124	G	110	G	83	G	92	G	109	G	103	G	33	G	38	G	31	G	34	G	26
57	B	137	B	208	B	125	B	36	B	45	B	42	B	80	B	110	B	85	B	64	B	69	B	82	B	91	B	43	B	49	B	41	B	45	B	36
49	R	62	R	186	R	167	R	27	R	35	R	32	R	85	R	116	R	89	R	80	R	91	R	92	R	73	R	20	R	44	R	25	R	33	R	18
55	G	64	G	194	G	188	G	37	G	42	G	37	G	101	G	137	G	110	G	98	G	110	G	106	G	88	G	25	G	48	G	29	G	37	G	22
65	B	78	B	202	B	202	B	53	B	50	B	45	B	94	B	125	B	80	B	74	B	87	B	88	B	80	B	36	B	57	B	39	B	48	B	33
43	R	47	R	48	R	193	R	34	R	29	R	37	R	80	R	119	R	92	R	92	R	76	R	63	R	82	R	40	R	24	R	27	R	35	R	23
48	G	51	G	56	G	207	G	44	G	36	G	42	G	97	G	141	G	110	G	111	G	95	G	78	G	97	G	45	G	28	G	31	G	39	G	27
58	B	64	B	65	B	216	B	61	B	47	B	48	B	86	B	129	B	86	B	89	B	73	B	56	B	84	B	56	B	39	B	42	B	49	B	38

BIG IDEA:

L'informazione è rappresentata in forma digitale



Operazioni sui bit

- Processore: decine/centinaia di miliardi di operazioni elementari su bit ogni secondo
- Poche e semplici istruzioni elementari («libretto delle istruzioni»): operatori logici (E, O, NON, O esclusivo... con 0 = falso e 1 = vero), semplici operazioni aritmetiche, trasferimento bit da/verso memoria, spostamenti di bit, confronti e salti ad altre istruzioni...
- Nessuna semantica (quella la diamo noi): solo circuiti che accendono/spengono interruttori

Dobbiamo astrarre

Es. Il nostro obiettivo è far stampare le iniziali di un utente

Inizi dell'informatica: inserire direttamente i bit che rappresentano le istruzioni del processore e i dati

Oggi:

```
N = input("Nome: ")
C = input("Cognome: ")
print(N[0], C[0])
```

Ma il processore sa sempre e solo fare quelle semplici istruzioni sui bit

Traduzione automatica in istruzioni macchina



Fondazione
I Lincei per la Scuola

Assembly, 3748 righe

Cython, 3 righe

```
N = input("Nome: ")
C = input("Cognome: ")
print(N[0], C[0])
```

```
385b:  be 05 00 00 00      mov     esi,0x5
3860:  e8 cb f8 ff ff      call    3130 <PyOS_snprintf@pl
3865:  31 c0                xor     eax,eax
3867:  8a 4c 05 00          mov     cl, BYTE PTR [rbp+rax*1
386b:  8a 14 03              mov     dl, BYTE PTR [rbx+rax*1
386e:  84 c9                test    cl,cl
3870:  75 0e                jne     3880 <__pyx_pymod_exec
3872:  83 ea 30              sub     edx,0x30
3875:  80 fa 09              cmp     dl,0x9
3878:  0f 87 8e 00 00 00     ja      390c <__pyx_pymod_exec
387e:  eb 0f                jmp     388f <__pyx_pymod_exec
3880:  38 d1                cmp     cl,dl
3882:  75 0b                jne     388f <__pyx_pymod_exec
3884:  48 f5 c0              inc     rcx
```

Macchina astratta

- Possiamo dimenticarci delle istruzioni macchina sui bit
- Macchina astratta (es. «macchina Python», «macchina Scratch»)
 - Sa eseguire le istruzioni Python o Scratch
- Che cosa sa fare la macchina astratta?
 - Di nuovo, poche e semplici cose

Eseguire istruzioni in sequenza

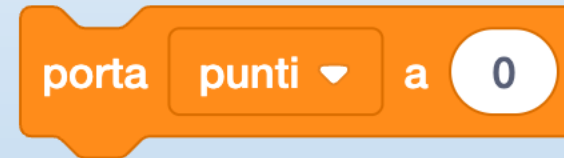
```
1 print("Qual è la risposta?")  
2 print(42)  
3 print("Ma qual era la domanda?")
```



Dare nomi a valori

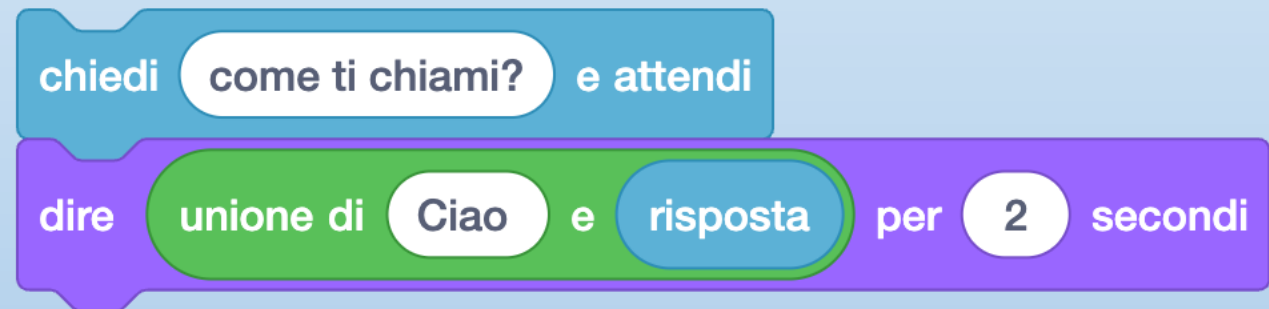
- Memorizzare i valori, organizzandoli in strutture più o meno semplici

```
1 a = 3
2 b = 4
3 c = 3*a + b**2
4 L = ['Michael', 'Lodi']
```



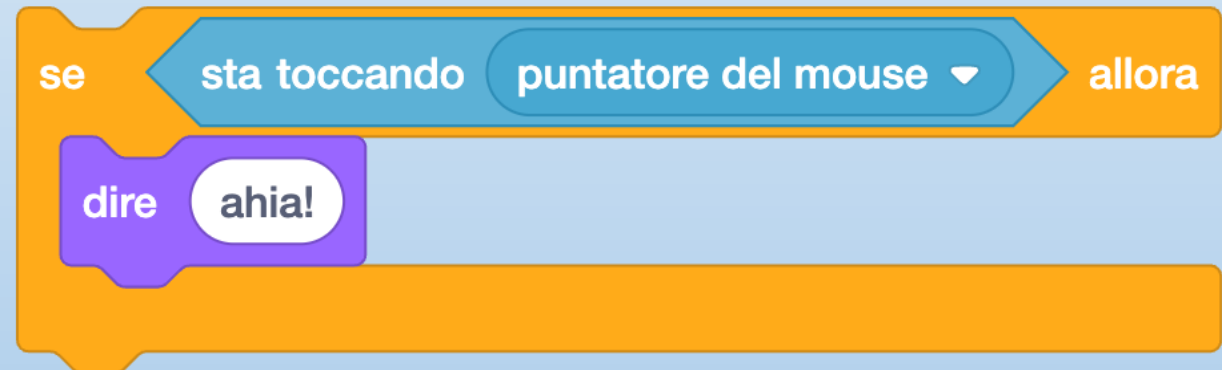
Prendere dati in input Stampare a schermo

```
1 n = input("Nome: ")
2 e = int(input("Eta: "))
3
4 print("Ciao", n, "di anni", e)
```



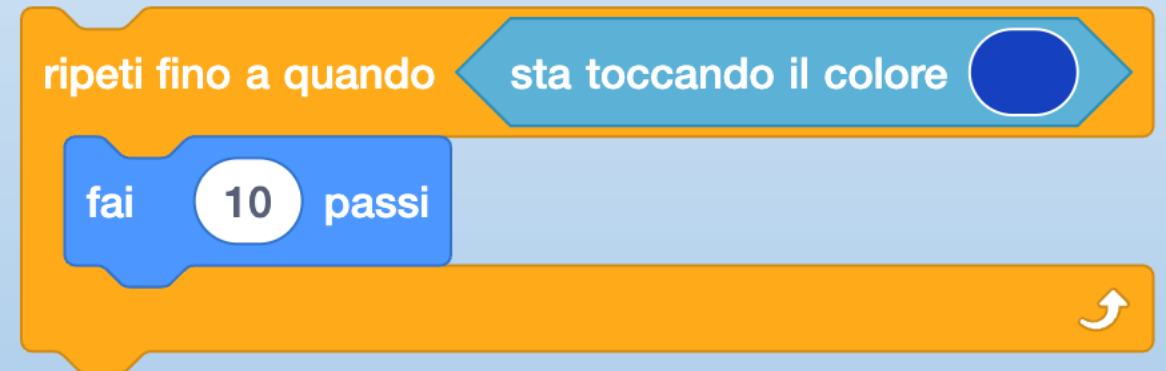
Prendere decisioni

```
if eta > 18:  
    print("Buon voto")  
else:  
    print("Ripassa più avanti")
```



Ripetere finché (non) si verifica una condizione

```
1 a = 72
2 b = 14
3
4 while b > 0:
5     a, b = b, a % b
6
7 print(a)
```



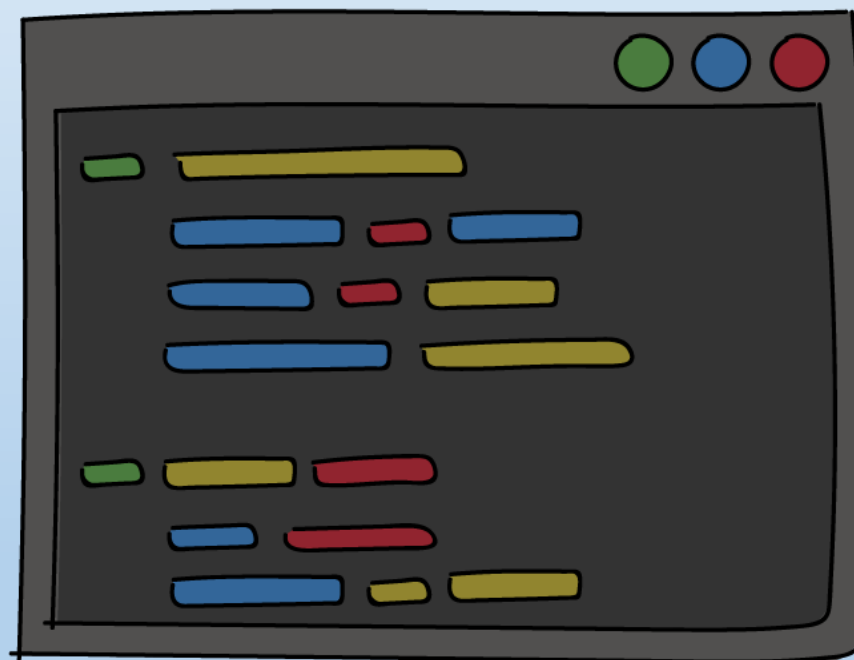
Universali



- Con queste sole istruzioni, possiamo scrivere qualsiasi* elaborazione

BIG IDEA:

I programmi rappresentano algoritmi e dati in una forma che può essere implementata su un computer



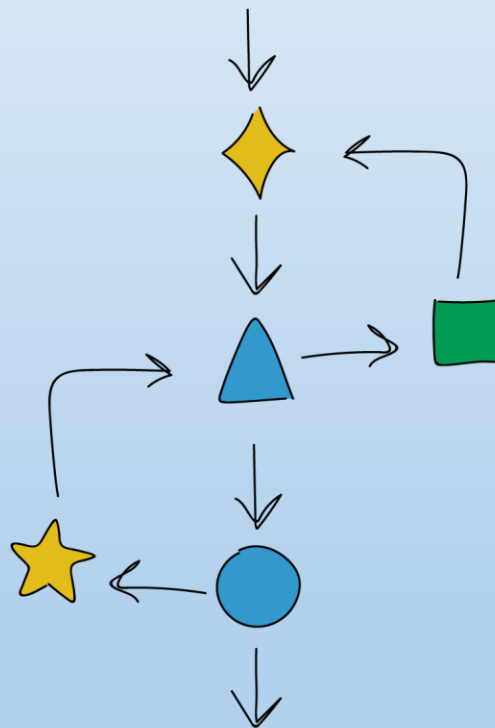
Problemi computazionali

- Classe di problemi (es. dati a e b trova x tale che $ax + b = 0$)
non specifico problema (trova x tale che $3x - 6 = 0$)
- Soluzione: un **algoritmo**, *una descrizione precisa e non ambigua di una procedura che si presta ad essere automatizzata*
 - *Informatica*: sempre in mente un preciso esecutore
- Programma: algoritmo scritto in un linguaggio di programmazione



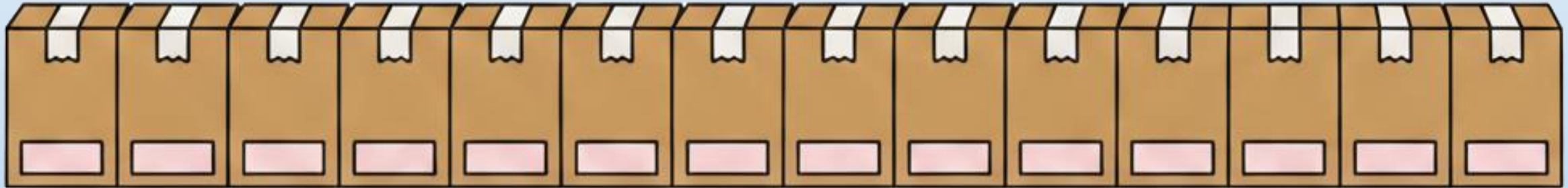
BIG IDEA:

Gli algoritmi interagiscono con i dati per risolvere problemi computazionali



Efficienza degli algoritmi

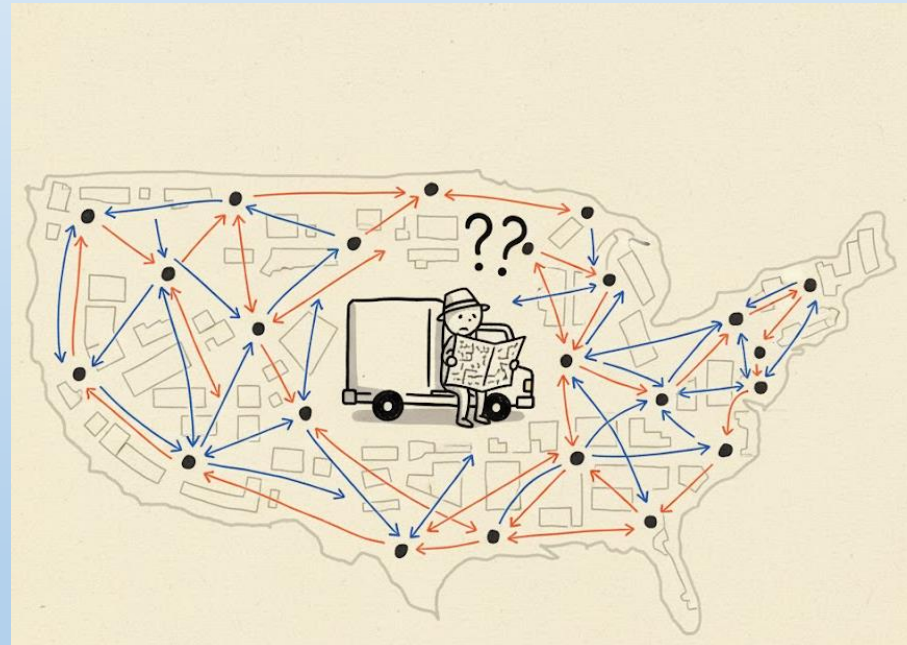
- Algoritmi diversi per risolvere lo stesso problema



- Ricerca lineare vs. dicotomica: miglioramento enorme!

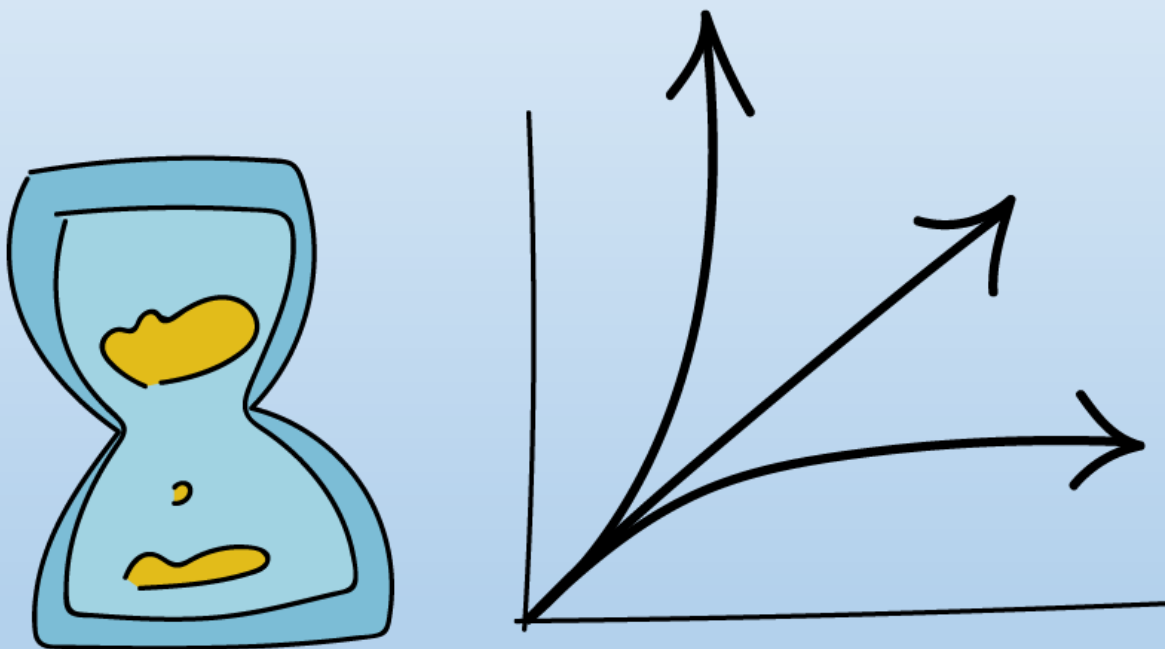
Problemi difficili

- Non abbiamo ancora trovato una soluzione «efficiente»
 - Forse non esiste!



BIG IDEA:

Le prestazioni degli algoritmi possono essere modellizzate e valutate



Problemi impossibili*

- Problemi computazionali - funzioni $f: \mathbb{N} \rightarrow \mathbb{N}$ (trasformano bit)
 - Tante quante i numeri reali $\mathbb{R}$
- Programmi: testi finiti su alfabeti finiti
 - Tanti quanti i numeri naturali $\mathbb{N}$
- Tantissimi problemi computazionali non sono *calcolabili*
 - Es. la correttezza di un programma



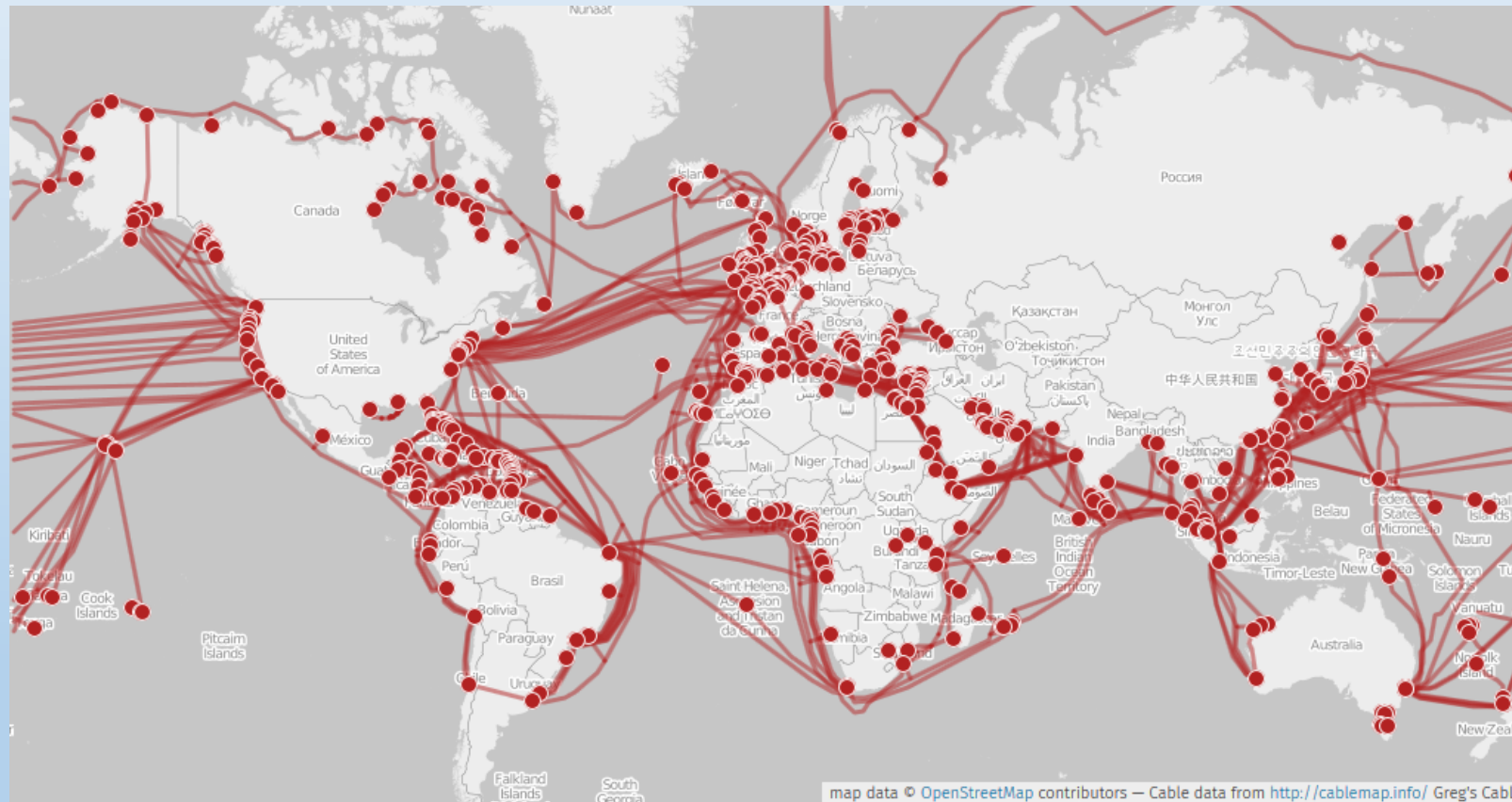
BIG IDEA:

Alcuni problemi computazionali non possono essere risolti con algoritmi

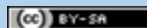


Trasmissione dell'informazione

- Internet: rete globale di computer



By cable data by Greg Mahiknecht, map by OpenStreetmap contributors -
<http://www.cablemap.info> (cable data by Greg Mahiknecht released under GPLv3)
<http://umap.openstreetmap.fr/de>, CC BY-SA 2.0,
<https://commons.wikimedia.org/wiki/index.php?curid=42437752>

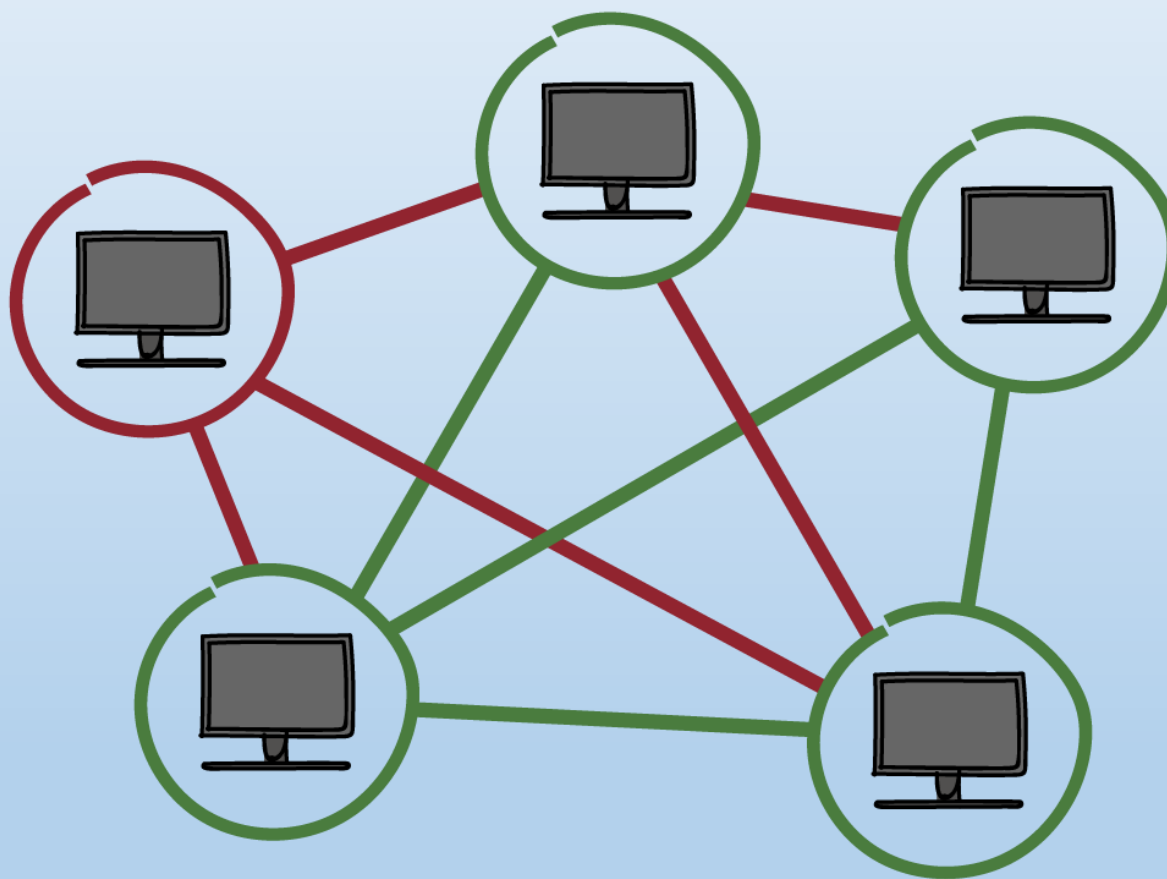


Michael Lodi. 2025. [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

Protocolli

- Regole di comunicazione standardizzate (HTTP, TCP/IP...)
- Dati divisi in piccole parti (*pacchetti*) e spediti a destinazione per strade potenzialmente diverse
- Il web è un servizio di Internet (documenti ipertestuali collegati...)

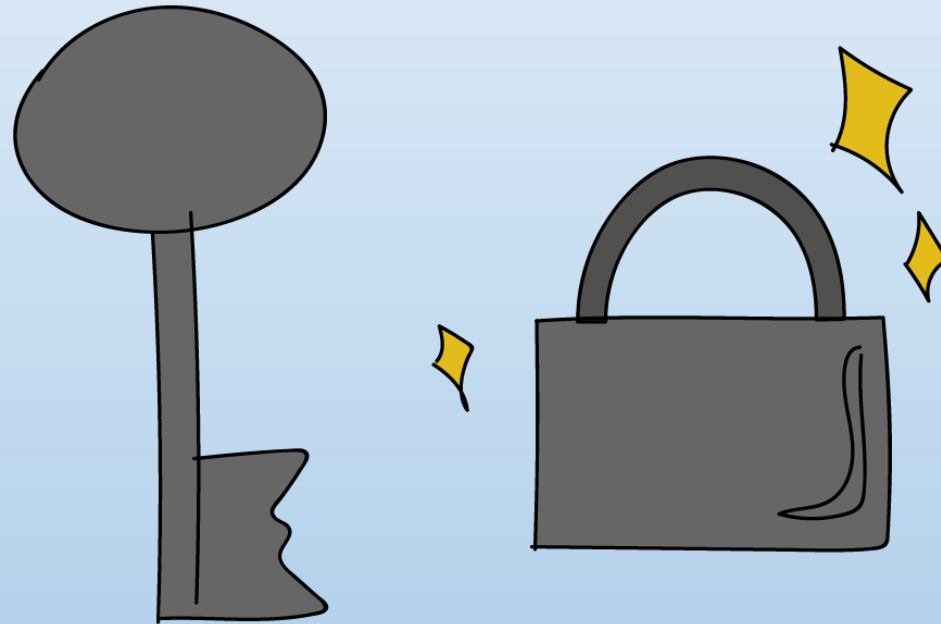
BIG IDEA: I sistemi digitali comunicano tra loro utilizzando protocolli



Crittografia

- Sicurezza non perfetta, ma «computazionale»
- Generare un segreto in modo sicuro, ma comunicando su un canale insicuro
 - Proprio usando quei problemi «difficili»

BIG IDEA: La protezione dei dati e delle risorse di sistema è fondamentale nei sistemi digitali

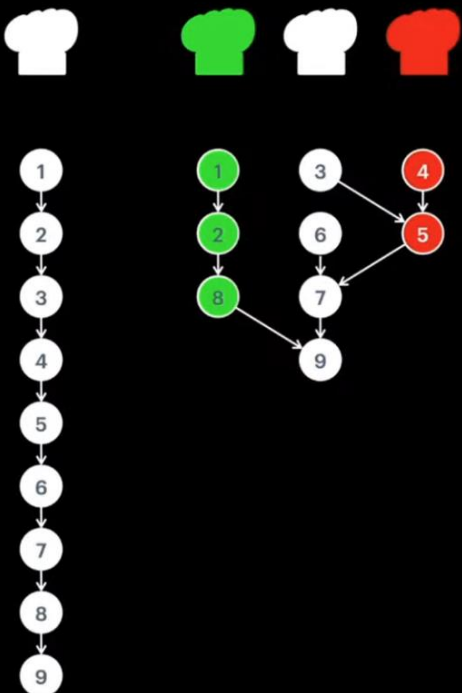


Calcolo parallelo e distribuito



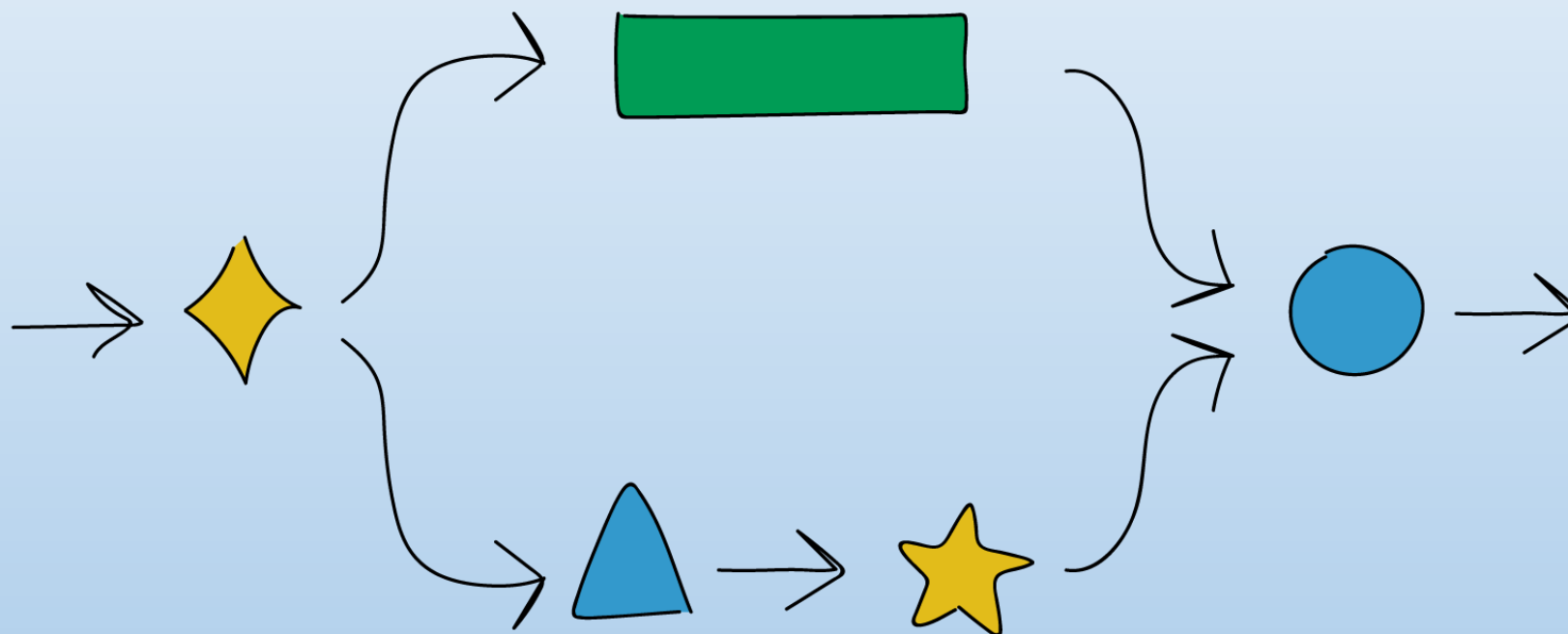
Tiramisù
Algoritmo parallelo

1. **Preparate il caffè**
2. **Inzuppate i savoiardi nel caffè**
3. Montate i tuorli delle uova insieme allo zucchero fino ad ottenere un bel composto chiaro e cremoso
4. **Lavorate poi il mascarpone con uno sbattitore fino a ottenere una crema senza grumi**
5. **Unite il composto di uova e zucchero preparato in precedenza**
6. Montate ora gli albumi a neve con un pizzico di sale, aggiungete la restante metà dello zucchero
7. Uniteli poco alla volta e delicatamente al composto di mascarpone e tuorli
8. **Adagiate in un contenitore i savoiardi e iniziate ad bagnarli col caffè**
9. Ricoprite i savoiardi inzuppati con uno strato di crema al mascarpone



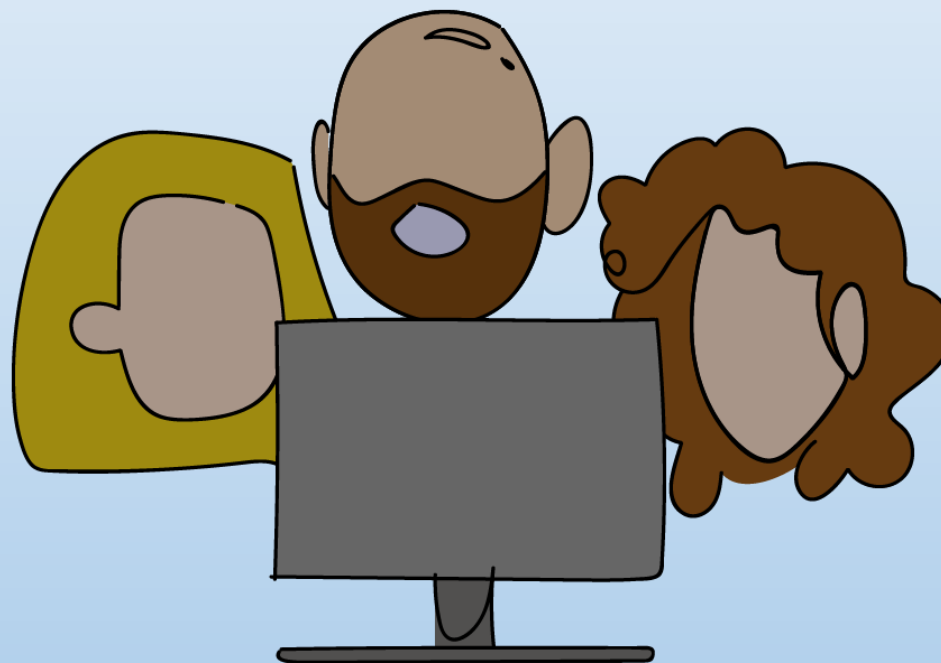
Conferenza generale DIGITALE – MARCO ALDINUCCI – 30/10/2025

BIG IDEA: Le operazioni dipendenti dal tempo nei sistemi digitali devono essere coordinate

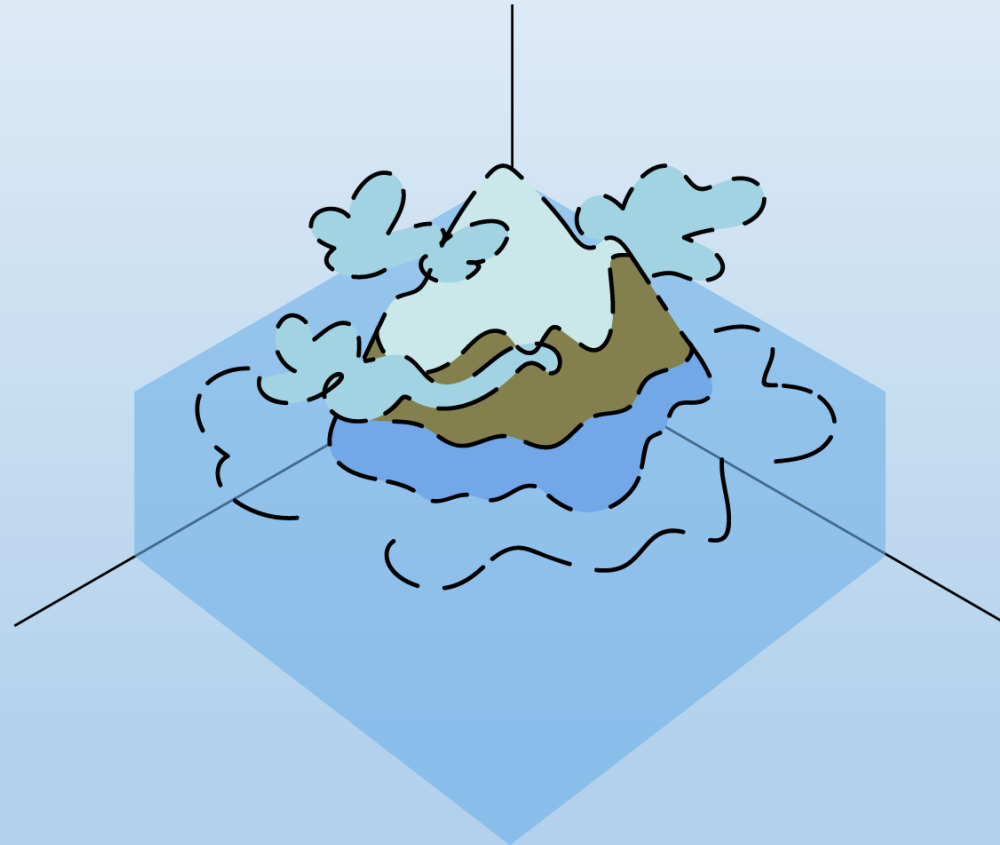


BIG IDEA:

I sistemi digitali sono progettati dagli umani
per soddisfare i bisogni umani



BIG IDEA: I sistemi digitali creano rappresentazioni virtuali di fenomeni naturali e artificiali



Le Big Ideas (dalla didattica delle scienze naturali)



- Le idee che gli studenti dovrebbero incontrare «nel corso dell'Educazione scientifica per comprendere, apprezzare e meravigliarsi del mondo naturale» [Harlen, 2015, ed. italiana]
- «Idee potenti»
- `lodi.ml/bigideas.pdf`

Le Big Ideas [Bell et. al, 2018]



- Non principi generali, aree disciplinari o argomenti curricolari, ma idee potenti e generali che catturano l'essenza di una disciplina
- Non una sequenza di argomenti, non per forza in sequenza
- Ciò che i giovani studenti dovrebbero "fare proprio" [sia per diventare informatici, sia per agire come cittadini del mondo digitale]
- Concetti che sono fondamentali di una disciplina, ma che non sono ovvi per il profano

Perché le Big Ideas

- Capire *perché* si insegnano certi contenuti
- fari per guidarci e non perderci nel mare di dettagli
 - i dettagli andranno insegnati, e sono fondamentali, ma senza «perdere la rotta»
- selezionare i concetti chiave (“conoscenza a lungo termine”) senza farsi prendere dall’ansia di **aggiungere nuovi contenuti** che emergono e cambiano rapidamente



Perché le Big Ideas

[Harlen, 2015, ed. italiana]



- curriculum scientifico troppo denso di contenuti e frammentario
- percezione da parte degli studenti della Scienza come una serie di informazioni disorganizzate e per loro poco significative.
- obiettivi della didattica non in termini di conoscenza di un insieme di fatti, fenomeni e teorie, ma in termini di una progressiva comprensione di idee chiave significative per la vita degli studenti sia durante che dopo il loro percorso scolastico.

Le grandi idee delle scienze naturali

[Harlen, 2015, ed. italiana]



1. Tutta la materia dell'Universo è costituita da particelle piccolissime
2. Gli oggetti agiscono a distanza su altri oggetti
3. Il movimento di un oggetto cambia quando su di esso agisce una forza
4. La quantità totale di energia nell'Universo è sempre la stessa ma una forma di energia può trasformarsi in un'altra
5. La composizione della Terra e della sua atmosfera e i processi che avvengono al loro interno "modellano" la superficie terrestre ed il clima
6. Il nostro sistema solare è una parte molto piccola di una tra i miliardi di galassie dell'Universo
7. Gli organismi sono costituiti da cellule ed hanno una vita limitata nel tempo
8. Gli organismi hanno bisogno di energia e di materia e per esse spesso stabiliscono relazioni di dipendenza o competizione con altri organismi
9. Le informazioni genetiche degli organismi si trasmettono da una generazione alla successiva
10. La diversità degli organismi, viventi ed estinti, è il risultato dell'evoluzione

Ogni idea ha una spiegazione

[Harlen, 2015, ed. italiana]



1 Tutta la materia dell'Universo è costituita da particelle piccolissime

Gli atomi sono i mattoni di tutta la materia, vivente e non vivente. Il comportamento e la disposizione degli atomi spiega le proprietà dei diversi materiali. Nelle reazioni chimiche gli atomi si riorganizzano per formare nuove sostanze. Ciascun atomo ha un nucleo che contiene neutroni e protoni, circondati da elettroni. Le cariche elettriche opposte dei protoni e degli elettroni si attraggono, mantenendo insieme gli atomi e determinando la formazione dei composti.

3 Il movimento di un oggetto cambia quando su di esso agisce una forza

La forza che agisce su un oggetto non è direttamente visibile, ma può essere individuata attraverso il suo effetto sul movimento o sulla forma dell'oggetto. Se l'oggetto non è in movimento le forze che agiscono su di esso sono uguali e contrarie: si equilibrano. Poiché la gravità agisce su tutti gli oggetti della Terra, quando un

Le grandi idee sulla scienza

[Harlen, 2015, ed. italiana]



- La Scienza si occupa di trovare la causa o le cause dei fenomeni del mondo naturale
- Spiegazioni scientifiche, teorie e modelli sono quelli che meglio si adattano alle evidenze disponibili al momento
- La conoscenza prodotta dalla Scienza trova applicazione nel campo dell'ingegneria e delle tecnologie
- Le applicazioni scientifiche hanno spesso implicazioni etiche, sociali, economiche e politiche

Le Big Ideas (dell'Informatica pre-universitaria) [Bell et. al, 2018]



1. L'informazione è rappresentata in forma digitale
2. Gli algoritmi interagiscono con i dati per risolvere problemi computazionali
3. Le prestazioni degli algoritmi possono essere modellizzate e valutate
4. I programmi rappresentano algoritmi e dati in una forma che può essere implementata su un computer
5. Le operazioni dipendenti dal tempo nei sistemi digitali devono essere coordinate
6. I sistemi digitali comunicano tra loro utilizzando protocolli
7. I sistemi digitali creano rappresentazioni virtuali di fenomeni naturali e artificiali
8. La protezione dei dati e delle risorse di sistema è fondamentale nei sistemi digitali
9. I sistemi digitali sono progettati dagli umani per soddisfare i bisogni umani
10. Alcuni problemi computazionali non possono essere risolti con algoritmi

Le «big ideas» della Crittografia

[Lodi et. al]



- I moderni sistemi crittografici sono basati sulla sicurezza computazionale, non sulla sicurezza perfetta
- È possibile generare un segreto condiviso in modo sicuro anche comunicando solo su un canale insicuro

Cinque grandi idee sull'Intelligenza Artificiale

5. IMPATTO SOCIALE

L'intelligenza artificiale può avere impatti sia positivi che negativi sulla società. Le tecnologie di IA stanno cambiando il modo in cui noi lavoriamo, viaggiamo, comunichiamo e ci prendiamo cura gli uni degli altri. Tuttavia, dobbiamo essere consapevoli dei danni che possono potenzialmente verificarsi. Per esempio, i pregiudizi nei dati utilizzati per addestrare un sistema di IA potrebbero portare alcune persone ad essere servite in modo peggiore rispetto ad altre. Perciò, è importante discutere degli impatti che l'IA sta generando nella nostra società e sviluppare criteri per il design etico e per la diffusione dei sistemi basati sull'IA.

4. INTERAZIONE NATURALE

Gli agenti intelligenti necessitano di molti tipi di informazioni per interagire in modo naturale con gli umani. Gli agenti devono essere in grado di conversare nel linguaggio umano, riconoscere le espressioni del viso e le emozioni, e avere un quadro generale sulla cultura e sulle convenzioni sociali per dedurre le intenzioni dai comportamenti osservati. Questi sono tutti problemi difficili da risolvere. I sistemi di IA odierni possono utilizzare parzialmente il linguaggio, ma mancano il ragionamento generale e le capacità di conversazione, paragonabili a quelle di un bambino.

1. PERCEZIONE

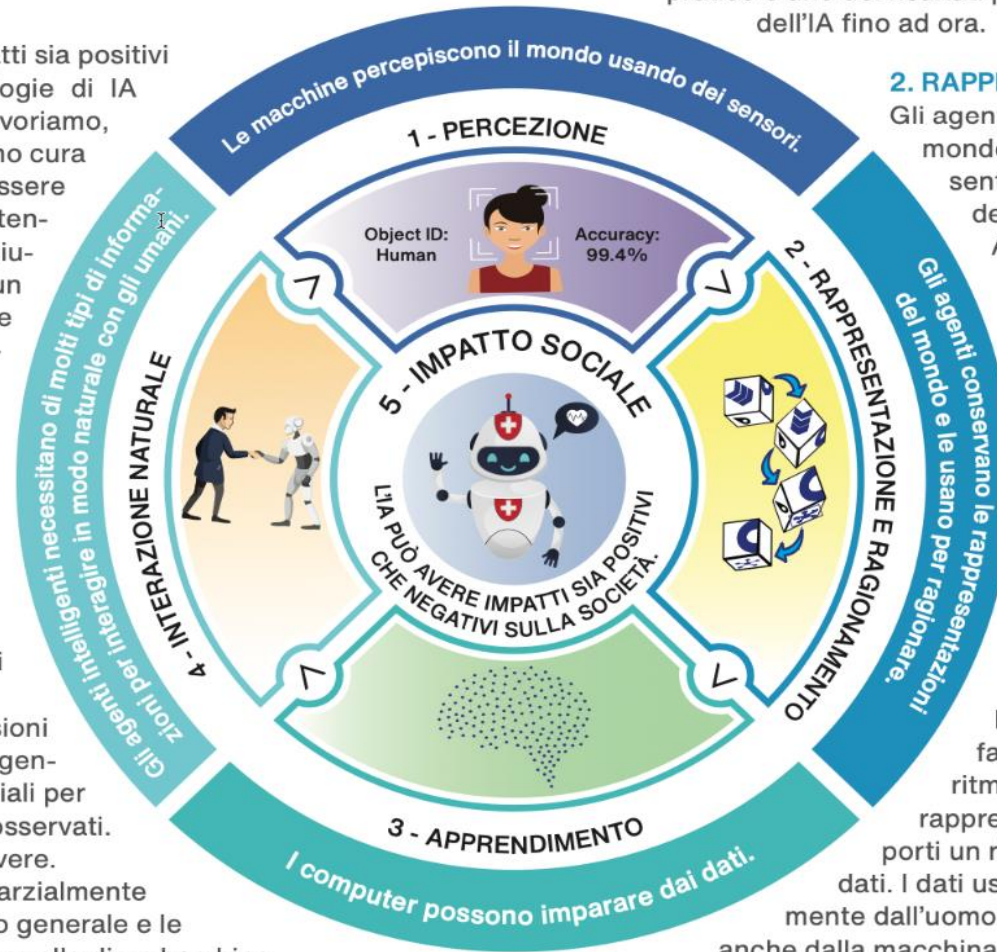
Le macchine percepiscono il mondo usando dei sensori. La percezione è il processo di estrazione del significato dai segnali sensoriali. Far "vedere" e "sentire" sufficientemente bene i computer per l'utilizzo pratico è uno dei risultati più significativi raggiunto nel campo dell'IA fino ad ora.

2. RAPPRESENTAZIONE E RAGIONAMENTO

Gli agenti conservano le rappresentazioni del mondo e le usano per ragionare. La rappresentazione è uno dei problemi fondamentali dell'intelligenza, sia naturale che artificiale. Attraverso le strutture dati, i computer costruiscono delle rappresentazioni che supportano gli algoritmi di ragionamento, i quali traggono nuove informazioni a partire da ciò che già si conosce. Sebbene gli agenti di IA siano in grado di ragionare su problemi molto complessi, non pensano nel modo in cui lo fa l'uomo.

3. APPRENDIMENTO

I computer possono imparare dai dati. L'apprendimento automatico (Machine Learning) è un tipo di deduzione statistica che riconosce degli schemi nei dati. Negli ultimi anni molti ambiti dell'IA hanno fatto progressi significativi grazie agli algoritmi di apprendimento che generano nuove rappresentazioni. Affinché questo approccio porti un risultato sono richieste enormi quantità di dati. I dati usati nell'addestramento sono forniti solitamente dall'uomo, ma in alcuni casi possono essere acquisiti anche dalla macchina stessa.



L'iniziativa IA per K-12 è un Progetto congiunto dell'associazione per il progresso dell'Intelligenza artificiale (AAAI) e l'associazione degli insegnanti di Computer Science (CTSA), finanziata dal premio DRL-1846073 della Fondazione National Science.



Questo lavoro è concesso con Licenza Creative Commons – Attribuzione – Non commerciale
Condividi allo Stesso Modo 4.0 Internazionale. Per leggere una copia della licenza visita:
<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.it>
Traduzione Ammagamma srl



Le Big Ideas della formazione HPC

[Barelli & Lodi, 2025]



Le *Big Ideas* per la didattica dell'HPC

Dai bisogni esistenti nella formazione
all'High-Performance Computing
alle raccomandazioni per la progettazione didattica



Michael Lodi. 2025. [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Big ideas dell'informatica



Tratte e tradotte e adattate da:

Tim Bell et al. The Big Ideas of K-12 Computer Science Education

Articolo accademico (in Inglese):

<http://bulletin.eatcs.org/index.php/beatcs/article/view/521>

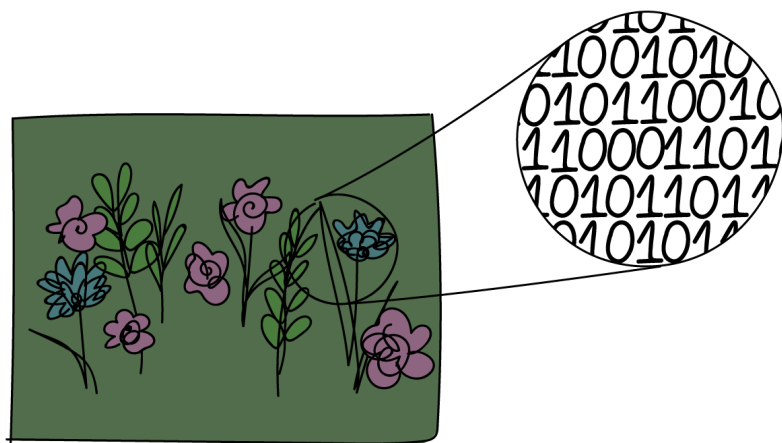
Testo per insegnanti con approfondimenti (in Inglese):

<https://www.csse.canterbury.ac.nz/big-ideas/BigIdeas-webdocument-15-Feb-2021.pdf>

Un video (al minutaggio corretto – ma tutto il video è interessante)
in cui racconto queste grandi idee:

<https://www.youtube.com/watch?v=QqjWMdR35XQ&t=3248s>

L'informazione è rappresentata in forma digitale



Tutta l'informazione (numeri, testo, immagini, suoni...) può essere rappresentata tramite (solitamente) due valori discreti

Questi valori sono scritti come cifre binarie (bit): rappresentano un mondo fisico "continuo" con valori discreti

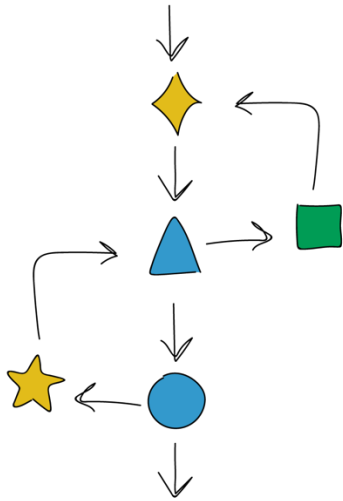
Anche le cifre sono un'astrazione: in realtà uso rappresentazioni elettriche, magnetiche, ottiche, ecc... in cui sia facile distinguere tra due "livelli"

Dispositivi **versatili** perché lo stesso hardware può essere usato per scopi molto diversi (differenza con dispositivi analogici, che sono specializzati)

Informazione in formato digitale può essere condivisa senza perdita di qualità

Si può modificare la modalità di memorizzazione dei dati, di modo che occupi meno spazio (compressione), per renderla priva di significato per gli intercettatori (crittografia) o per renderla utile anche se alcuni bit sono rovinati (controllo degli errori).

Gli algoritmi interagiscono con i dati per risolvere problemi computazionali



Un algoritmo può includere solo passi “atomici” che un computer “convenzionale” può svolgere autonomamente senza inferenze

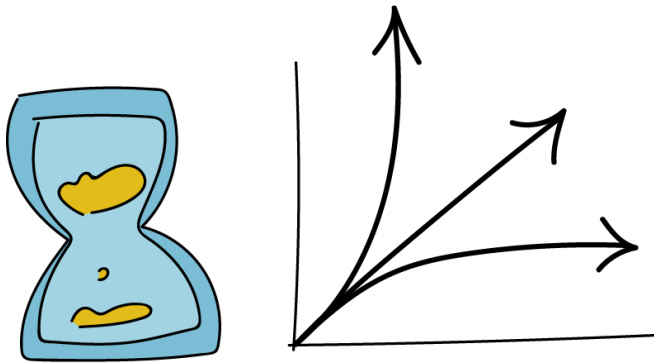
Usando solo tre strutture di controllo (sequenza, selezione, iterazione) e capacità di input, output, memorizzazione – o le equivalenti di queste 6 in altri modelli – può essere risolto ogni problema computazionale

Gli algoritmi sono una rappresentazione statica e finita che descrive un processo dinamico potenzialmente infinito

Un problema computazionale può avere diverse soluzioni (algoritmi), e un algoritmo è un **processo generale** che risolve tale problema per tutte le sue istanze (es. trovare il massimo di una lista di elementi), in contrasto con altre tipologie di problem solving umano

Ci sono diversi modi in cui memorizzare i dati (strutture dati) che influenzano es. la velocità di accesso e gli algoritmi utilizzabili (es. trovare il massimo in una lista ordinata o disordinata)

Le prestazioni degli algoritmi possono essere modellizzate e valutate



Le risorse principali che un algoritmo utilizza sono il tempo di esecuzione e lo spazio di memoria.

Un algoritmo inefficiente porta anche a sprecare energia e ha quindi un impatto ambientale (oltre che sulla batteria dello smartphone)

Solitamente la complessità non è misurata precisamente (dipendente dall'hardware), ma come un tasso di crescita, es. come funzione della dimensione dell'input (indipendente dall'hw)

Il tempo impiegato per risolvere un problema con un algoritmo non è sempre proporzionale alla dimensione dell'input; a volte è meglio di così, e a volte è (molto) peggio

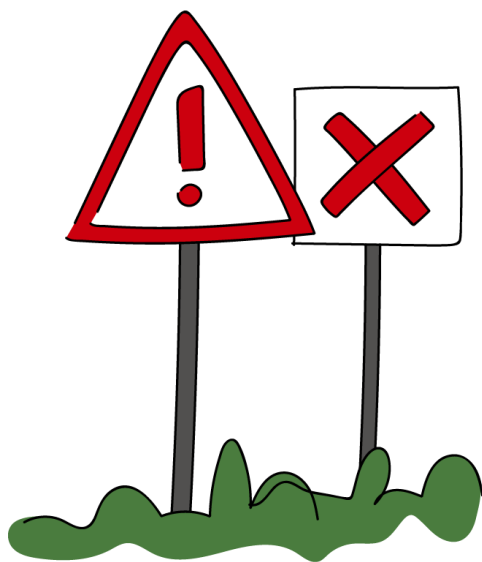
Ci possono essere algoritmi più o meno efficienti per uno stesso problema

Per alcuni problemi importanti (es. strada migliore per un corriere) conosciamo solo algoritmi con tempi di fatto intrattabili (es. esponenziali, infattibili anche con hw più veloci) e non sappiamo se esistono algoritmi migliori

A volte possiamo sfruttarli a nostro vantaggio (es. crittografia)

Possiamo considerare euristiche che danno soluzioni approssimate

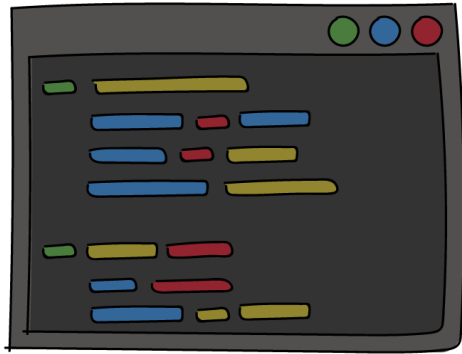
Alcuni problemi computazionali non possono essere risolti con algoritmi



Ci sono alcuni problemi computazionali per i quali possiamo dimostrare che non esiste un algoritmo che li risolve (problemi non calcolabili / indecidibili)

Alcuni sarebbero molto utili, anche agli insegnanti: es. un programma che ci dice se un altro programma termina correttamente o meno per ogni input (o decidere se due programmi sono equivalenti)

I programmi rappresentano algoritmi e dati in una forma che può essere implementata su un computer



Programmare vuol dire trasformare algoritmi in istruzioni che possono essere eseguite da un computer, scritte in un linguaggio di programmazione precisamente definito

Per descrivere qualunque computazione bastano tre strutture di controllo (sequenza, selezione, iterazione) e tre strumenti gestire i dati (input, output, memorizzazione)

Esistono altri paradigmi di programmazione che utilizzano strutture diverse ma egualmente espressive (es. ricorsione)

Differenze tra i linguaggi hanno a che fare soprattutto con i diversi scopi specifici di un linguaggio

Linguaggi semplici come Scratch sono tanto espressivi quanto altri linguaggi avanzati quali Python o C

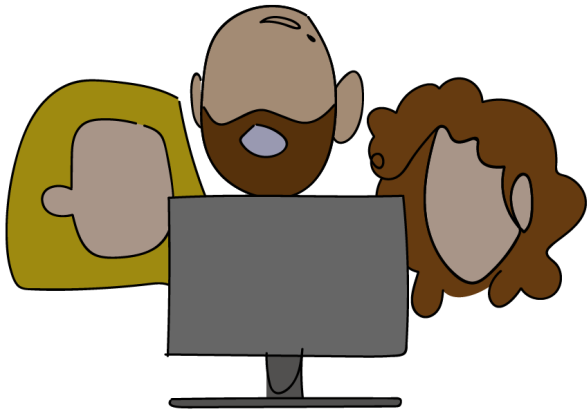
Programmare è un atto creativo, visto che esistono diversi modi per risolvere un problema

Programmare non è solo fare “coding”, è anche design, test e debug, ecc.

I programmi sono memorizzati, come tutti gli altri dati, con cifre binarie all'interno del computer

Esistono linguaggi formali meno espressivi per compiti più semplici (es. controllare il formato di una espressione, definire la sintassi dei linguaggi di programmazione...)

I sistemi digitali sono progettati dagli umani per soddisfare i bisogni umani



I dispositivi digitali devono essere veloci, affidabili e soddisfare le esigenze delle persone che li utilizzano

Tre aree importanti, che richiedono di considerare aspetti psicologici, sociologici e fisiologici sono

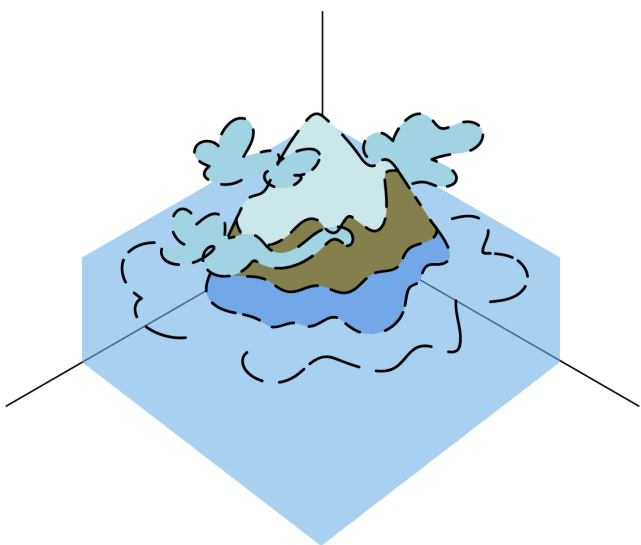
- l'interazione persona-computer, per la creazione di interfacce semplici e usabili
- l'ingegneria del software, per sviluppare software su larga scala in modo collaborativo
- l'etica, che considera l'impatto di una nuova tecnologia sull'uomo, le responsabilità di chi ci lavora e se dovremmo o meno costruirla

I programmi non sono scritti per le macchine (nemmeno il sorgente, che è scritto per altri esseri umani)

Meglio pensare a **“spiegare a un altro essere umano cosa vogliamo che il computer faccia”** (Knuth, 1984)

Nella società digitale sono necessarie le competenze per prendere decisioni informate sui temi come voto, **privacy**, proprietà intellettuale, automazione del lavoro, impatti ambientali

I sistemi digitali creano rappresentazioni virtuali di fenomeni naturali e artificiali



Le simulazioni e i sistemi virtuali vengono utilizzati per creare versioni virtuali (modelli) dei processi nel mondo fisico, o anche per creare scenari immaginari, per ridurre costi e rischi (simulare prima di costruire o di porsi in situazioni pericolose), o per dare l'impressione di essere in un modo immaginario

La simulazione si usa nella scienza quando teoria e sperimentazione non bastano

Oggi si può utilizzare anche la ricerca di pattern in grandi moli di dati

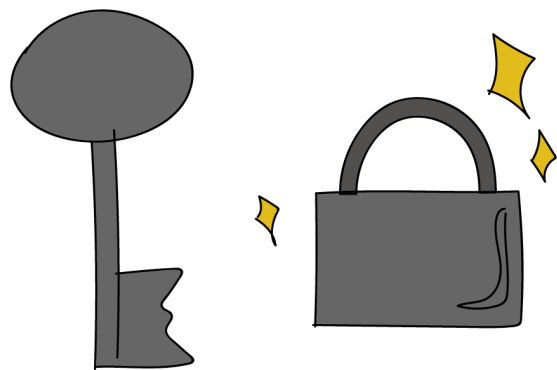
Le macchine virtuali sono simulazioni di computer all'interno di computer, per questioni di sicurezza (es. protezione dell'hw) e flessibilità

L'intelligenza artificiale modella l'intelligenza umana, ma è ancora un programma per computer soggetto alle sue limitazioni

Le decisioni delle IA "sub-simboliche" possono non essere intelleggibili dall'uomo

...

La protezione dei dati e delle risorse di sistema è fondamentale nei sistemi digitali



L'anello più debole nella sicurezza di un sistema è l'utente, che deve quindi essere consapevole

La sicurezza include la riservatezza (impedire accesso non autorizzato alle informazioni), la disponibilità (gli utenti legittimi possono accedere alle loro informazioni) e l'integrità (le informazioni sono accurate).

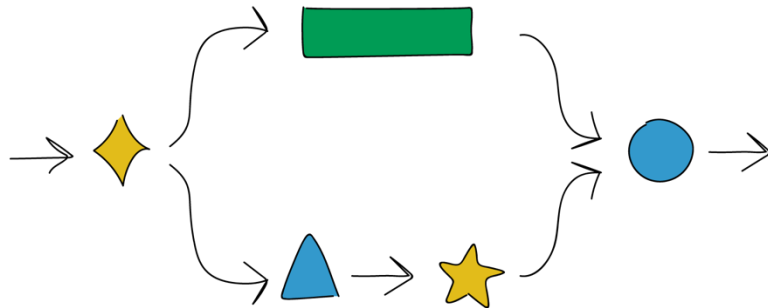
Conoscere come funziona la crittografia a chiave pubblica **può far sentire gli utenti più sicuri online e utilizzare più consciamente pagamenti elettronici, identità e firme digitali, voto elettronico, ecc.**

Gli utenti devono conoscere le tecniche che possono utilizzare, quali crittografia, rilevazione e blocco di attacchi, autenticazione, tecniche di recovery...

Sistemi di password ben fatti non memorizzano le password (ma gli hash), e sono comunque in grado di dire se abbiamo inserito quella corretta

...

Le operazioni dipendenti dal tempo nei sistemi digitali devono essere coordinate



Le dipendenze tra operazioni in parallelo può limitare la capacità di suddividere un problema in più parti eseguibili da componenti indipendenti

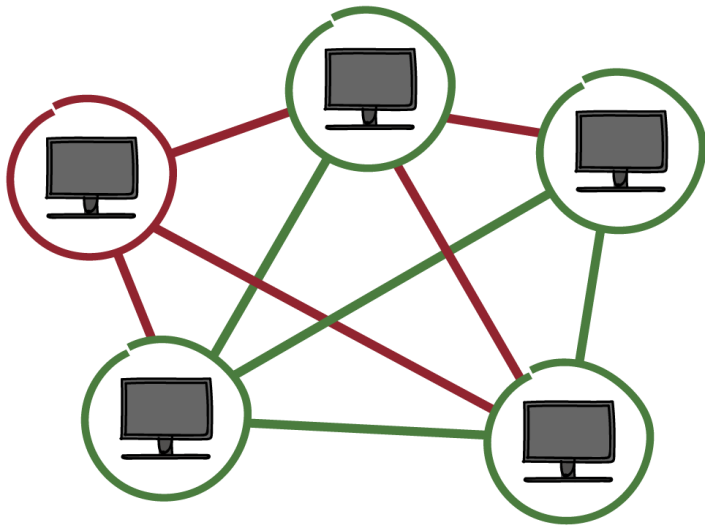
Raddoppiare il processori non sempre dimezza il tempo di esecuzione

I sistemi digitali hanno molti componenti che possono funzionare indipendentemente; questi componenti possono funzionare in parallelo e con schedule diversi

La computazione può essere distribuita tra diversi processori, anche remoti, che devono essere coordinati

Processi concorrenti devono essere progettati attentamente per evitare race condition (dove l'ordine in cui si verificano gli eventi determina un diverso risultato), deadlock (in cui due processi sono entrambi in attesa l'uno dell'altro) e starvation (in cui alcune operazioni impediscono ad altri di ottenere l'accesso alle risorse).

I sistemi digitali comunicano tra loro utilizzando protocolli



I protocolli assicurano che i dati vengano scambiati in modo sicuro ed efficiente

I protocolli di comunicazione in genere suddividono i dati in pacchetti

Questi protocolli sono spesso espressi in "livelli", dai dettagli dei connettori fisici, fino a dettagli come la suddivisione dei dati in pacchetti, la numerazione di essi e il riconoscimento del loro ricevimento, e al più alto livello, specificano come un programma può interfacciarsi con un programma su un altro computer.

I protocolli sono alla base di un'ampia gamma di servizi come Internet (es. TCP/IP), il web (es. HTTP), l'e-mail (es. IMAP), comunicazione tra dispositivi (es. Bluetooth)

I protocolli sono creati dall'uomo e sono soggetti a negoziazione, tramite standard formali o de-facto

...

Bibliografia



Tutto su `lodi.ml/infonin`

[Harlen, 2015] W. Harlen “Working with Big Ideas of Science Education”, Inter Academy Panel, 2015. Edizione italiana curata da A. Pascucci

[Bell et al., 2018] Tim Bell, Paul Tymann, and Amiram Yehudai (2018) The Big Ideas in Computer Science for K-12 Curricula. Bulletin of EATCS 1, 124 (2018).

[AIK12.org, 2020] https://ai4k12.org/wp-content/uploads/2021/08/AI4K12_Five_Big_Ideas_Poster_Italian.pdf

[Lodi, 2024] Michael, L. (2024). You can securely generate a shared secret by communicating only on an insecure channel. Zenodo.

[Barelli & Lodi, 2025] Eleonora Barelli e Michael Lodi (2025). Le Big Ideas della formazione HPC: Dai bisogni esistenti nella formazione in High-Performance Computing alle raccomandazioni per la progettazione didattica. Zenodo.

[Crippa et al., 2025a] Marta Crippa, Violetta Lonati, Luca Lamanna, Laura Branchetti. (2025) Educare al pensiero algoritmico: riflessioni e proposte per il primo ciclo. ITADINFO 2025.

[Crippa et al., 2025b] Marta Crippa, Laura Branchetti, Luca Lamanna e Violetta Lonati (2025). Conoscenza procedurale, algoritmi e problemi: spunti di riflessione da una prospettiva interdisciplinare tra matematica e informatica. XXXIX Convegno Nazionale “Incontri con la Matematica”.

Grazie!

Michael Lodi

*Dipartimento di Informatica – Scienza e Ingegneria
Alma Mater Studiorum – Università di Bologna*

michael.lodi@unibo.it
lodi.ml