

# Wrap-up

Informatica di base – a.a. 2018/2019

Silvio Peroni

[0000-0003-0530-4305](mailto:silvio.peroni@unibo.it)

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia  
[silvio.peroni@unibo.it](mailto:silvio.peroni@unibo.it) – [@essepuntato](https://www.unibo.it/sitoweb/silvio.peroni/) – <https://www.unibo.it/sitoweb/silvio.peroni/>

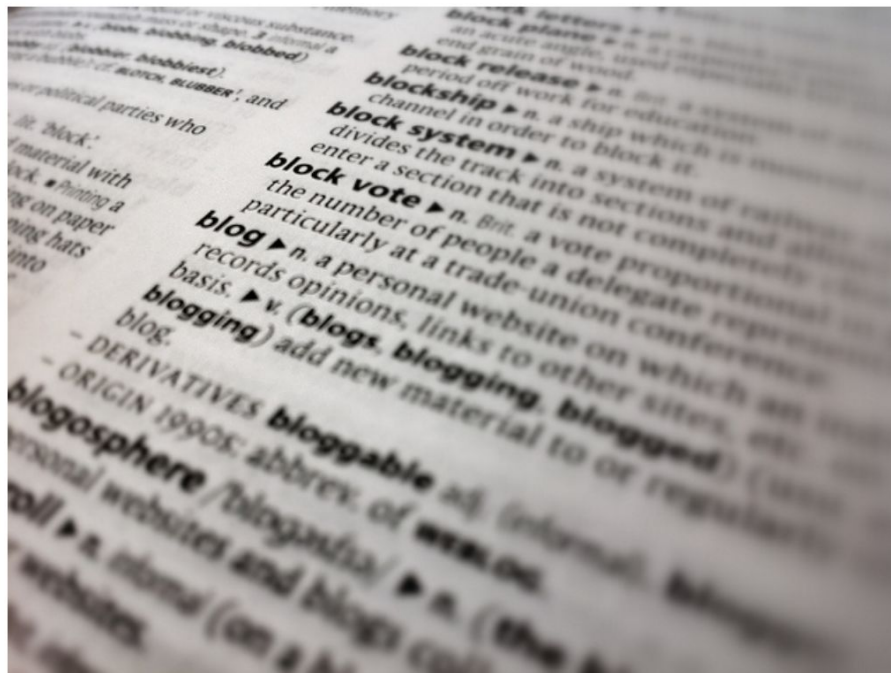


Quest'opera è distribuita con [Licenza Creative Commons Attribuzione 4.0 Internazionale](https://creativecommons.org/licenses/by/4.0/)



# Dizionario

Una collezione non ordinata di elementi definiti da coppie chiave-valore che si possono contare, dove la chiave non è ripetibile



								
USD	GBP	CAD	CHF	AUD	INR	TND	AED	JPY
1.17926	0.89380	1.51150	1.16817	1.56403	76.5506	2.93654	4.33138	132.582

# Approssimazione della soluzione di Turing

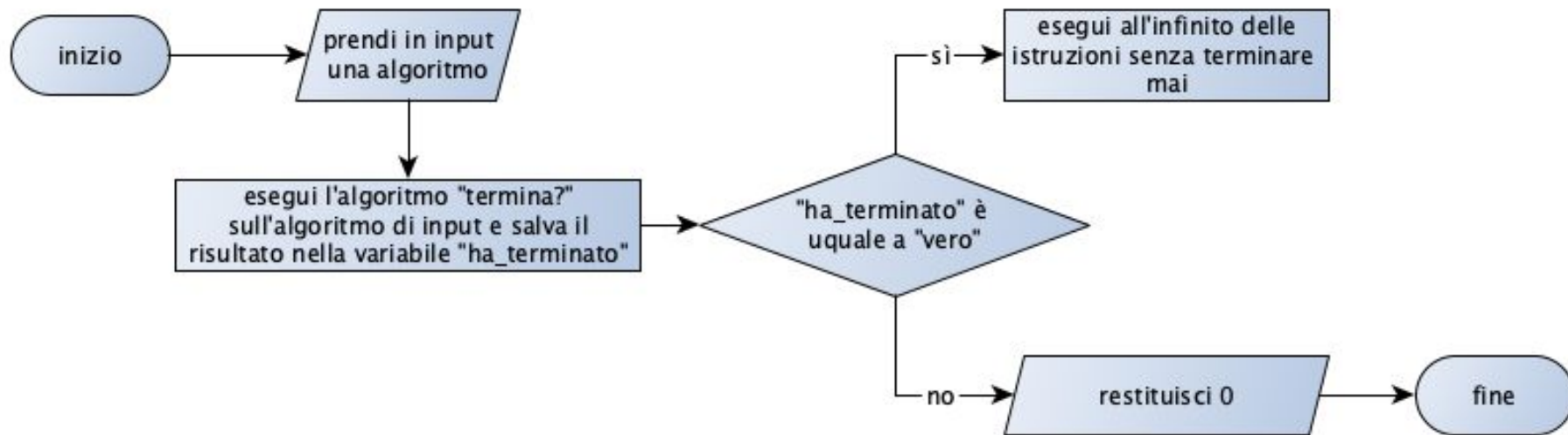
Premessa: supponiamo **sia possibile sviluppare** l'algoritmo "termina?", che prende in **input un certo algoritmo** e restituisce "**vero**" nel caso in cui l'algoritmo specificato come **input termina**, mentre restituisce "**falso**" **in caso contrario**

NB: è soltanto un algoritmo ipotetico, stiamo supponendo che possiamo svilupparlo in qualche modo, senza mostrare come farlo davvero

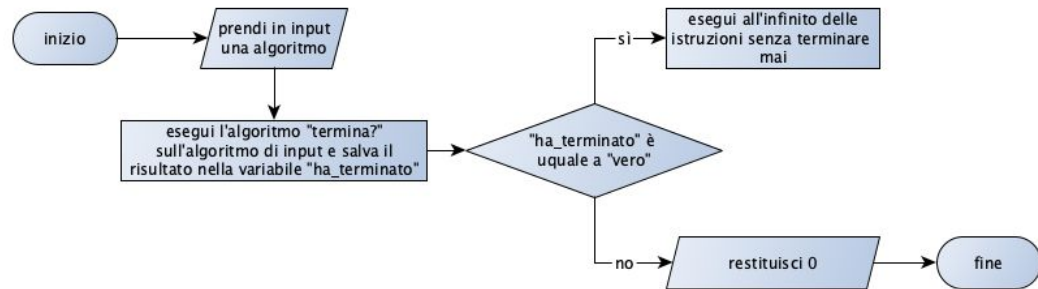
Usiamo questo algoritmo per crearne un altro

# Nuovo algoritmo

Nuovo algoritmo: prendi in input un algoritmo e restituisci 0 se l'algoritmo in input termina, mentre non terminare in caso contrario



# E se usiamo il nuovo algoritmo come suo input?



Caso 1: se l'algoritmo "termina?" afferma che il nuovo algoritmo **termina**, conseguentemente (per come è definito) il nuovo algoritmo **non termina** l'esecuzione

Caso 2: se l'algoritmo "termina?" afferma che il nuovo algoritmo **non termina**, e conseguentemente (per come è definito) il nuovo algoritmo restituisce 0 e **termina** l'esecuzione

Paradosso: l'algoritmo che verifica se un altro termina **non può esistere**

# Grammatica formale

Un insieme di regole di produzione di forma  $\text{premessa} ::= \text{espressione}$ ,  
dove la  $\text{premessa}$  e l' $\text{espressione}$  possono contenere uno o più:

- simboli **terminali** (specificati tra virgolette), che identificano tutti i simboli elementari del linguaggio in considerazione (come nomi, verbi, etc.)
- simboli **non terminali** (specificati tra parentesi angolari), che identificano tutti i simboli di una grammatica formale che possono essere sostituiti da una combinazione di simboli terminali e non terminali

Esempio:

$\langle \text{frase} \rangle ::= \text{"Io"} \langle \text{verbo} \rangle$

$\langle \text{verbo} \rangle ::= \text{"scrivo"} \mid \text{"leggo"}$

Il simbolo non terminale  $\langle \text{frase} \rangle$  è il simbolo iniziale.

Fraasi che possiamo produrre da questa grammatica:

- Io scrivo
- Io leggo

# Gerarchia di Chomsky

Chomsky ha proposto una gerarchia per descrivere formalmente le relazioni che possono esistere tra diverse grammatiche in termini delle loro possibili strutture sintattiche che sono in grado di generare

Queste tipologie sono caratterizzate dal tipo di simboli che possono essere usati nella premessa e nell'espressione delle regole di produzione (lettere greche usate per indicare una combinazione di simboli terminali e/o non terminali):

- grammatiche **regolari** –  $\langle nt \rangle ::= "t" \mid "t" \langle nt \rangle$
- grammatiche **libere dal contesto** –  $\langle nt \rangle ::= \gamma$
- grammatiche **dipendenti dal contesto** –  $\alpha \langle nt \rangle \beta ::= \alpha \gamma \beta$
- grammatiche **ricorsivamente enumerabili** –  $\alpha ::= \beta$

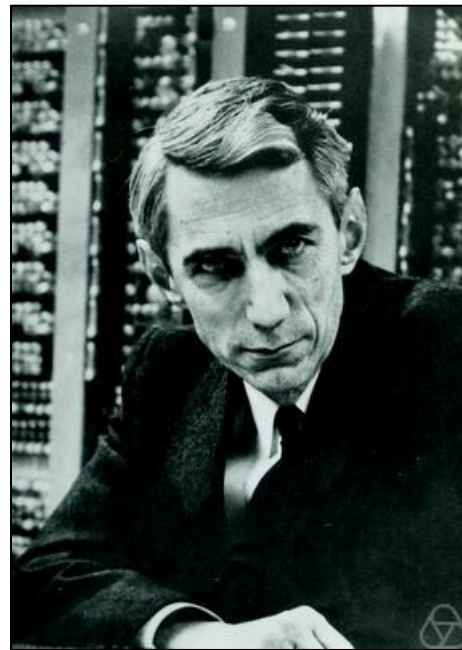
# Come sono espressi questi limiti

**Byte:** unità minima di informazione occupabile su un computer, che storicamente corrisponde al numero di bit necessari

**Bit** (contrattura di *binary digit*): l'unità minima di informazione che si può scambiare in una comunicazione e può assumere solo uno di due valori: 0 o 1

$$1 \text{ byte} = 8 \text{ bit}$$

Il concetto di bit è stato usato da diversi studiosi del passato (ad esempio Babbage con le schede perforate), ma introdotto come termine formalmente da Claude Shannon (1948)





# Teoria dell'informazione

Nell'articolo che introduce il bit, Shannon in realtà mette le basi di uno specifico campo di ricerca e studio chiamato **teoria dell'informazione**

La teoria dell'informazione si occupa di studiare come quantificare, memorizzare, e scambiare informazione, che ha tuttora svariate applicazioni pratiche, oltre che aver caratterizzato e veicolato l'invenzione di diverse tecnologie del passato, incluso lo sviluppo di Internet

Esempio: evincere la capacità massima di un canale per trasmettere informazione in modo affidabile è stato uno degli studi più importanti della teoria dell'informazione (a cui l'MTU è strettamente connesso)

# Da bit a simboli e viceversa

Ogni carattere, numero, programma, applicazione all'interno di un computer, così come un qualunque messaggio da scambiare in Internet, è **codificato** come una sequenza di bit che, in qualche modo, vengono poi **decodificati** con la sequenza di caratteri corretta dalla particolare applicazione che riceve ed interpreta quell'informazione

Per esempio, nella codifica binaria tradizionale dei numeri interi, il numero 0 è rappresentato dalla sequenza "0", il numero 1 da "1", il numero 2 da "10", il numero 3 da "11", il numero 4 da "100", e così via

# Gli indirizzi IP (v4)

Sono definiti mediante l'uso di **4 byte**, uno per ogni numero

Ogni byte, ovvero 8 bit, permette di definire un **numero intero da 0 a 255**

Codifica / decodifica binaria: ognuno degli otto bit di un numero facente parte dell'indirizzo IP ha assegnato uno specifico valore – il valore 1 è associato al bit più a destra, mentre a quello successivo verso sinistra è assegnato un valore doppio rispetto al precedente (in questo caso 2), e così via

<b>indirizzo IP</b>	208	.	80	.	154	.	224
<b>bit</b>	1 1 0 1 0 0 0 0		0 1 0 1 0 0 0 0		1 0 0 1 1 0 1 0		1 1 1 0 0 0 0 0
<b>valore</b>	128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1		128 64 32 16 8 4 2 1
<b>somma</b>	128+64 +16		64 +16		128 +16 +8 +2		128+64+32

# Esame

La prova d'esame è composta da una prova soltanto, da sostenere a computer in uno dei laboratori informatici a disposizione

Ogni prova è composta da 33 domande a risposta multipla. Ogni domanda risposta correttamente vale 1 punto, mentre una risposta sbagliata o assente vale 0 punti – esiste una sola risposta corretta per domanda

A fine prova, viene restituito il risultato a video che corrisponde al voto finale

Prossimi appelli (max. 7 turni di un'ora ciascuno, dalle 9:00):

17/5/2019 Aula D

17/6/2019 Laboratorio Informatico via Zamboni 34

15/7/2019 Laboratorio Informatico via Zamboni 34

# Domande

Le domande verteranno su tutto il contenuto del corso a disposizione sulla piattaforma IOL – che di fatto è lo stesso dei PDF pubblicati

Le domande sono organizzate in due “tipologie” diverse:

- 19 domande *teoriche*, che verteranno sulle conoscenze del contenuto del corso e mirano ad indagare quanto lo studente ha appreso del materiale messo a disposizione
- 14 domande di *ragionamento*, in cui si valuta la capacità di pensiero computazionale dello studente nel rispondere a particolari situazioni

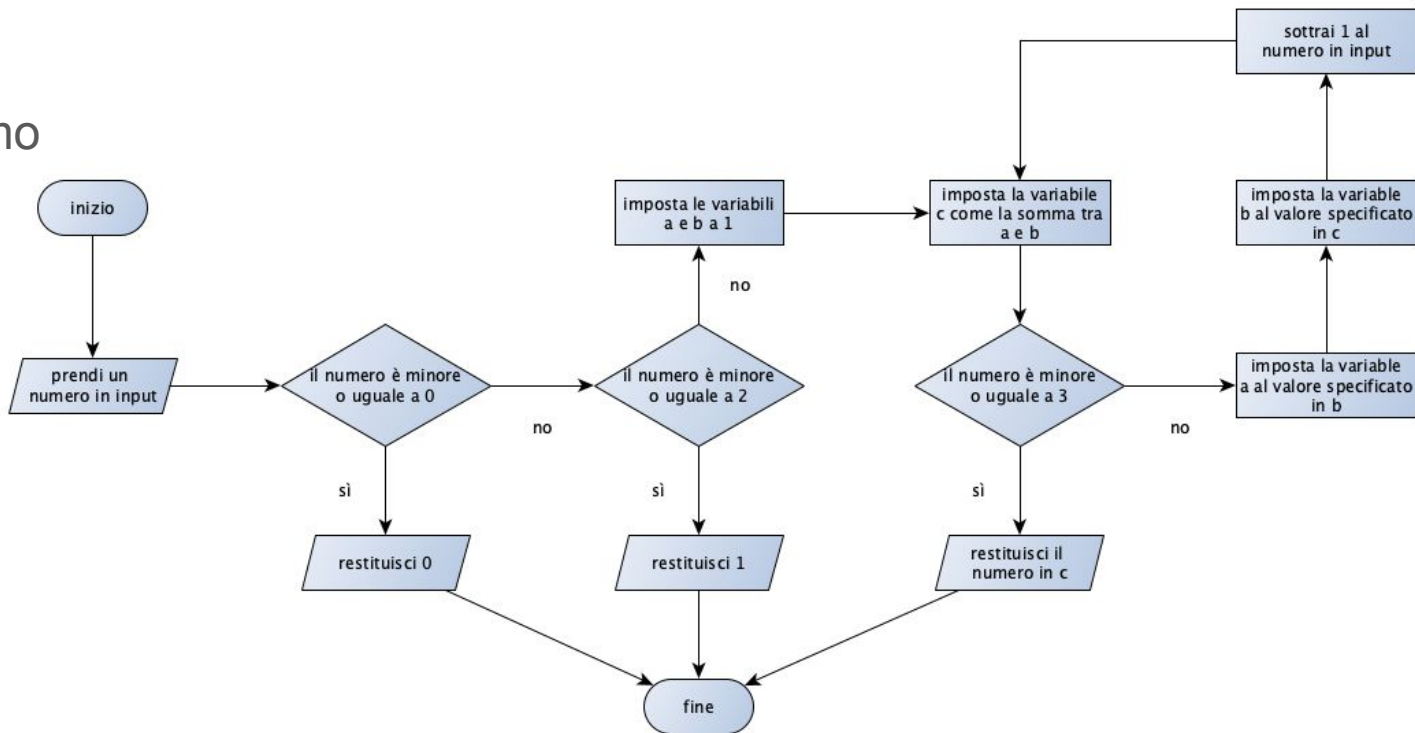
# Esempio domanda teorica

Che cos'è un programma?

- ❑ È un documento scritto in un linguaggio informale che è solitamente usato per comunicare i passi principali di un algoritmo ad un umano
- ❑ È la specifica (o implementazione), fatta da parte di un programmatore, di un certo algoritmo usando un particolare linguaggio di programmazione comprensibile da computer elettronico
- ❑ È l'astrazione di una procedura passo passo che prende qualcosa come input e produce un certo output
- ❑ È un particolare dispositivo hardware, parte di un computer elettronico, che permette di eseguire degli algoritmi a partire da specifici input

# Esempio di domanda di ragionamento

Cosa viene restituito dal seguente algoritmo se lo si esegue specificando il numero “4” come input?



- ☐ 1
- ☐ 10
- ☐ 3
- ☐ 5

# Wrap-up

## Le tecnologie informatiche nelle scienze umane

Informatica di base – a.a. 2018/2019

Silvio Peroni

[0000-0003-0530-4305](https://orcid.org/0000-0003-0530-4305)

Dipartimento di Filologia Classica e Italianistica, Università di Bologna, Bologna, Italia

[silvio.peroni@unibo.it](mailto:silvio.peroni@unibo.it) – [@essepuntato](https://www.essepuntato.it) – <https://www.unibo.it/sitoweb/silvio.peroni/>



Quest'opera è distribuita con [Licenza Creative Commons Attribuzione 4.0 Internazionale](https://creativecommons.org/licenses/by/4.0/)

