

ALGORITMI ED ESECUTORI

Fondamenti di Programmazione 2022/2023

Francesco Tortorella



Che cosa si intende per INFORMATICA ?

- Scienza della rappresentazione e dell'elaborazione dell'informazione
 - L'informazione è il concetto principale dell'Informatica.
 - L'elaborazione dell'informazione che si considera è quella svolta in maniera automatica tramite un calcolatore (o computer)
 - Per poter essere automatizzata, l'elaborazione deve essere formalizzata in maniera sistematica e rigorosa



Che cosa si intende per INFORMATICA ?

- Scienza dell'astrazione
 - creare il giusto **modello** per un problema e individuare le tecniche appropriate per risolverlo in modo **automatico**
 - L'obiettivo è quello di sostituire una situazione del mondo reale complessa e particolareggiata con un modello comprensibile e privo di dettagli inessenziali, all'interno del quale si possa risolvere il problema



Che cosa si intende per INFORMATICA ?

- Mettiamo insieme le due definizioni:
Obiettivo dell'Informatica è creare delle astrazioni di problemi del mondo reale che possano essere rappresentate ed elaborate all'interno di un sistema di calcolo al fine di eseguire dei procedimenti risolutivi in modo automatico



Esempio

- Si devono dividere lastre di marmo di Carrara, rettangolari e di dimensioni $A \times B$ (con A e B variabili) in tanti quadrati uguali avente il lato della maggiore lunghezza possibile e senza generare sfrido.
- Si supponga che tutte le dimensioni siano numeri interi.



Esempio

- Si vuole realizzare un procedimento automatico che, per ogni lastra, verifichi se ciò è possibile, date le dimensioni, e, in caso positivo, fornisca la lunghezza del lato del quadrato (può essere diverso per ogni lastra).



Esempio

- Quali sono gli aspetti importanti del problema ?
- Quali sono i dati a disposizione ?
- Quali sono i dati richiesti ?

- Come si può ridefinire in **forma astratta** il problema?



Esempio

- Quali sono gli aspetti importanti del problema ?
- Quali sono i dati a disposizione ?
- Quali sono i dati richiesti ?

- Come si può ridefinire in **forma astratta** il problema?



Esempio

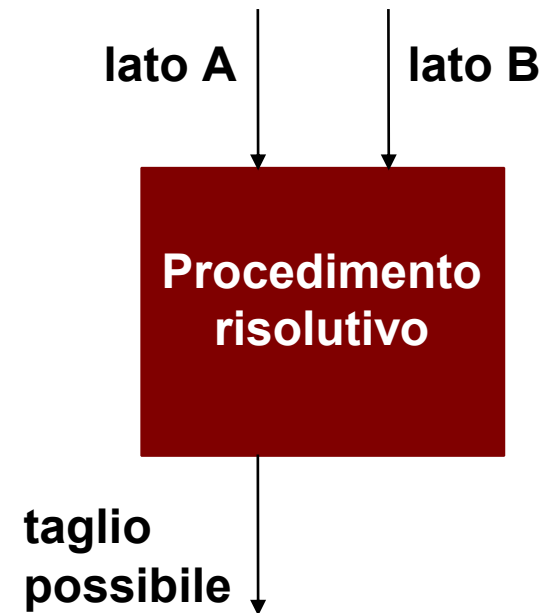
- Quali sono gli aspetti importanti del problema ?
- Quali sono i dati a disposizione ?
- Quali sono i dati richiesti ?

- Come si può ridefinire in **forma astratta** il problema?



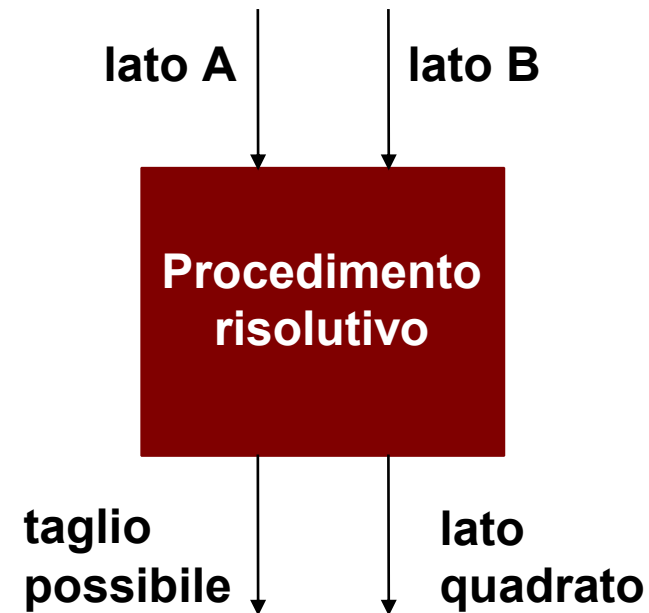
Esempio

- Quali sono gli aspetti importanti del problema ?
 - Quali sono i dati a disposizione ?
 - Quali sono i dati richiesti ?
-
- Come si può ridefinire in **forma astratta** il problema?



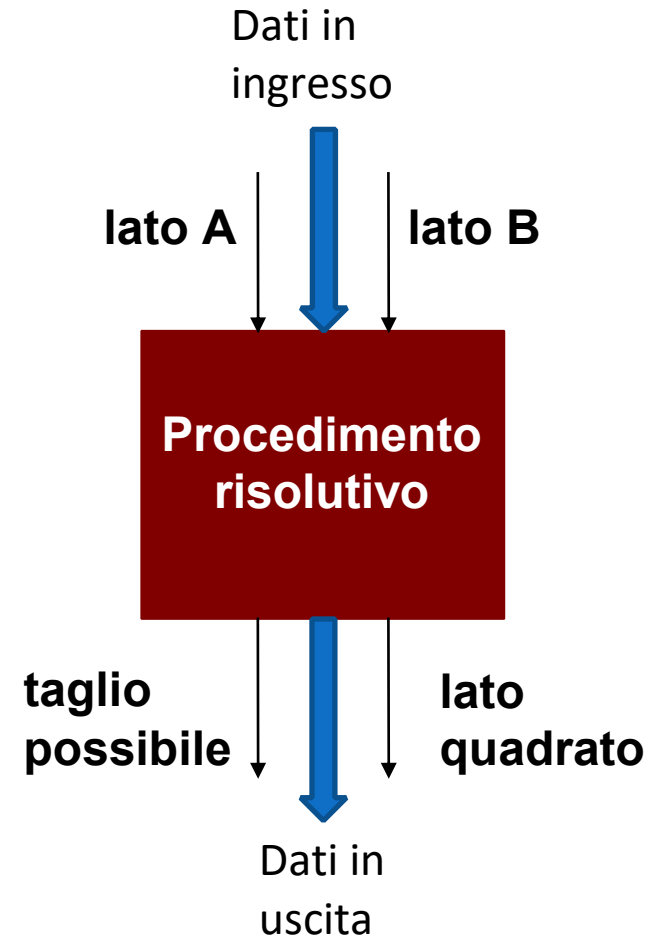
Esempio

- Quali sono gli aspetti importanti del problema ?
 - Quali sono i dati a disposizione ?
 - Quali sono i dati richiesti ?
-
- Come si può ridefinire in **forma astratta** il problema?



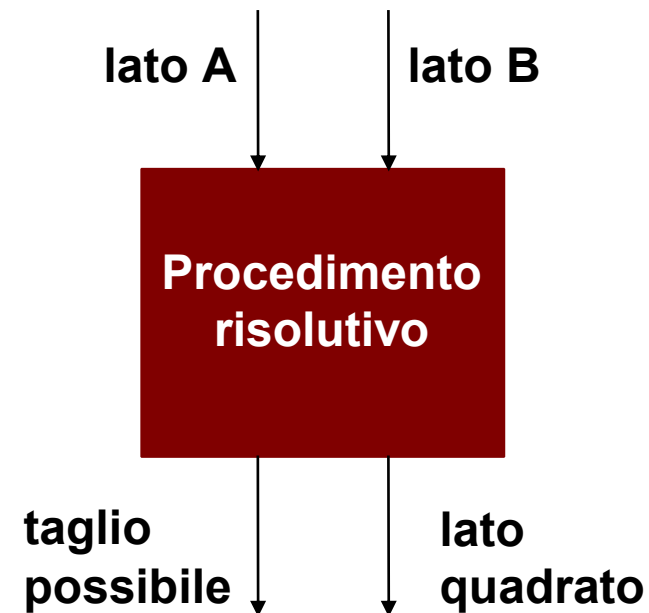
Esempio

- Quali sono gli aspetti importanti del problema ?
 - Quali sono i dati a disposizione ?
 - Quali sono i dati richiesti ?
-
- Come si può ridefinire in **forma astratta** il problema?



Ridefiniamo il problema

- Calcolare il Massimo Comune Divisore (MCD) dei due numeri interi A e B
- Se il MCD è diverso da 1, il taglio è possibile e la misura del lato del quadrato è data dal MCD.
- Se il MCD è uguale a 1, il taglio non è possibile



Come realizzare il procedimento risolutivo?

- Dovremo specificare il procedimento in modo che sia eseguibile in maniera automatica.
- Quali caratteristiche dovrebbe avere?
 - Dovrebbe essere una sequenza finita di operazioni chiaramente definite
 - Dovrebbe essere specificato l'ordine di esecuzione di ogni operazione
 - Dovrebbe essere chiaramente specificata la condizione di termine della sequenza



Che cosa fa?

1. Leggi due numeri X e Y , con $X > Y$
2. Dividi X per Y e ottieni il resto R
3. Se $R=0$, termina: il risultato è Y
4. Sostituisci X con Y
5. Sostituisci Y con R
6. Torna al punto 2.



Proviamo ...

X	Y	R	Output
96	36		

X	Y	R	Output
83	17		



Algoritmo di Euclide

- Uno degli algoritmi più antichi conosciuti per il calcolo del MCD, presente negli Elementi di Euclide intorno al 300 a.C. (proposizione VII.2)
- Probabilmente non scoperto da Euclide. Certamente era conosciuto da Eudosso di Cnido intorno al 375 a.C.
- Aristotele (intorno al 330 a.C.) ne ha fatto cenno ne I topici, 158b, 29-35.
- Non richiede la fattorizzazione dei due interi.



Che cos'è un algoritmo?

- E' un procedimento **sistematico**, costituito da un insieme **finito** di operazioni, ognuna delle quali è **precisa** (non ambigua) ed **eseguibile**, da applicare ai **dati in ingresso** perché possa fornire dei **dati in uscita**.



Caratteristiche interessanti ...

- L'algoritmo è del tutto generale, ma, in qualsiasi caso specifico, il procedimento avrà termine e fornirà una risposta precisa in un numero finito di passi.
- A ogni passo, è perfettamente chiaro quale operazione si debba compiere e anche la decisione circa il momento in cui il procedimento si debba ritenere concluso è perfettamente definita.



Caratteristiche interessanti ...

- Con il termine passo intendiamo un'operazione di natura qualsiasi, ma elementare e determinata
- Con più passi si possono costruire procedimenti più articolati che realizzano un particolare compito.
- Con **finitezza** si indica un aspetto fondamentale: perché sia svolto in maniera utile il compito non può richiedere un numero infinito di passi.

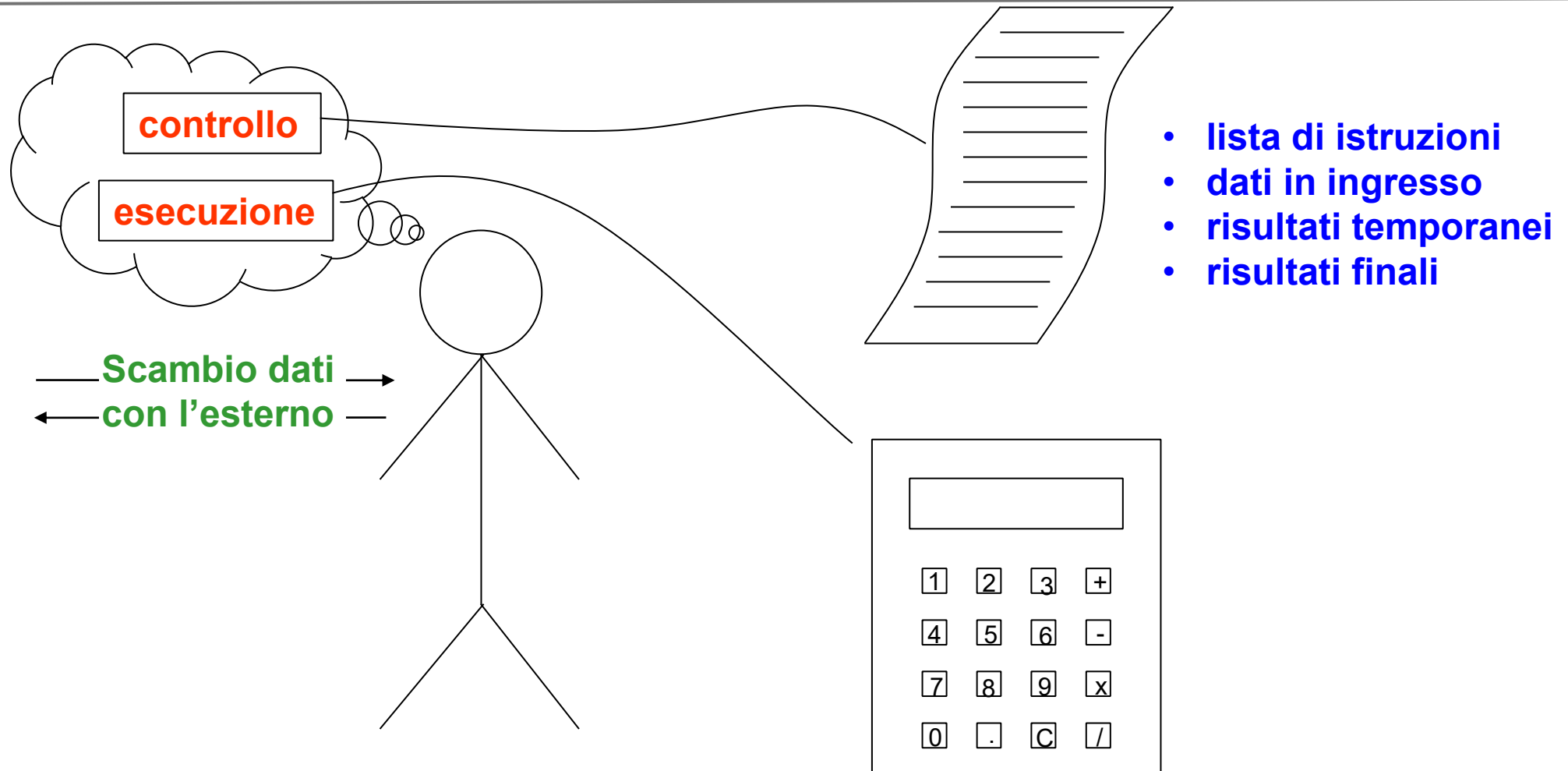


Chi esegue l'algoritmo?

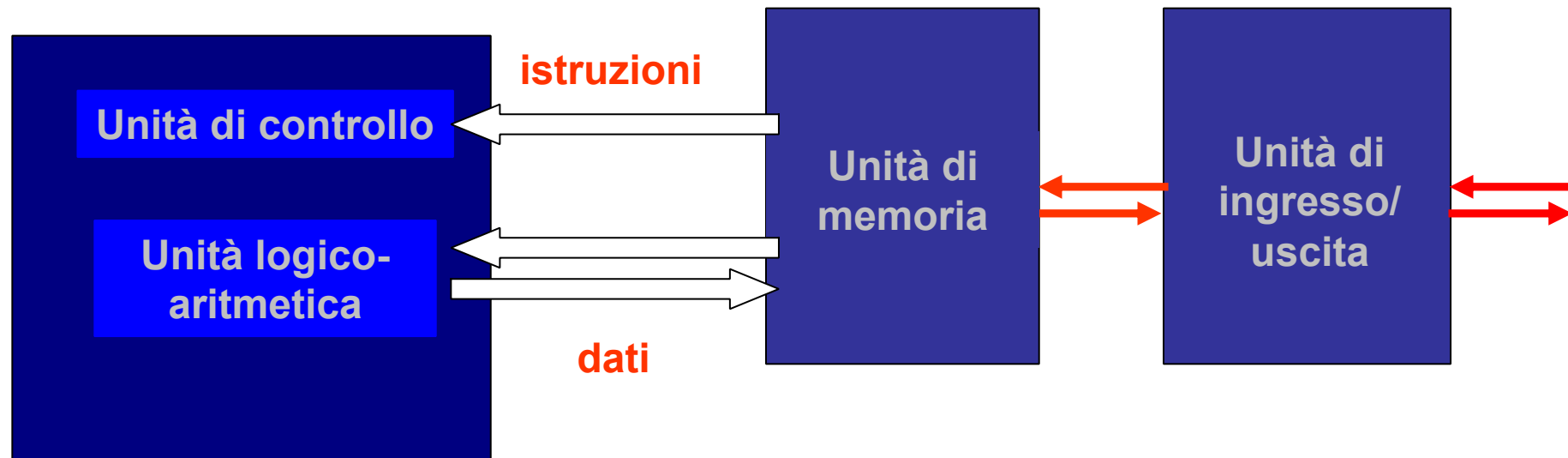
- Una volta definito, l'algoritmo deve essere sottoposto ad un **esecutore**.
- L'esecutore deve essere in grado di:
 - **interpretare** correttamente la sequenza di comandi
 - **eseguire** ognuno dei comandi forniti
 - **memorizzare** informazioni su opportuni supporti che permettano di **accedere** alle informazioni memorizzate e **modificarle**
- **Questione:**
 - **l'esecutore deve essere consapevole di quello che sta facendo?**



Un esecutore «umano»



Un esecutore artificiale

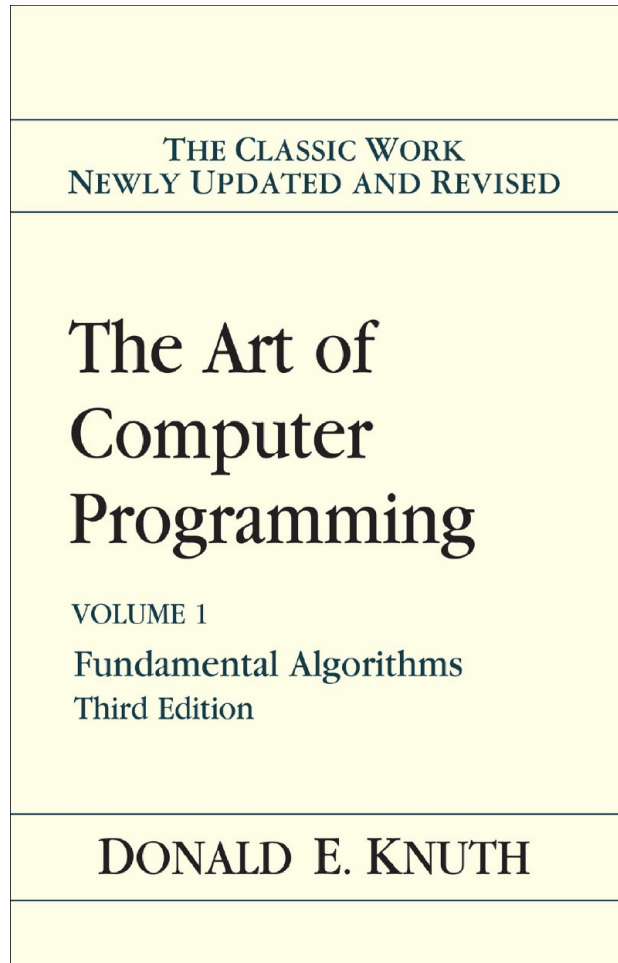


Algoritmo e programma

- Differenze tra i due tipi di esecutori (umano vs artificiale):
 - rappresentazione delle istruzioni
 - rappresentazioni dei dati
- La descrizione di un algoritmo è indipendente dall'esecutore che dovrà eseguirlo
- Di conseguenza, è necessario rappresentare istruzioni e dati in un formato tale che l'esecutore sia capace di memorizzare e manipolare



Algoritmo e programma



Algorithm E (*Euclid's algorithm*). Given two positive integers m and n , find their *greatest common divisor*, that is, the largest positive integer that evenly divides both m and n .

E1. [Find remainder.] Divide m by n and let r be the remainder. (We will have $0 \leq r < n$.)

E2. [Is it zero?] If $r = 0$, the algorithm terminates; n is the answer.

E3. [Reduce.] Set $m \leftarrow n$, $n \leftarrow r$, and go back to step E1. ■

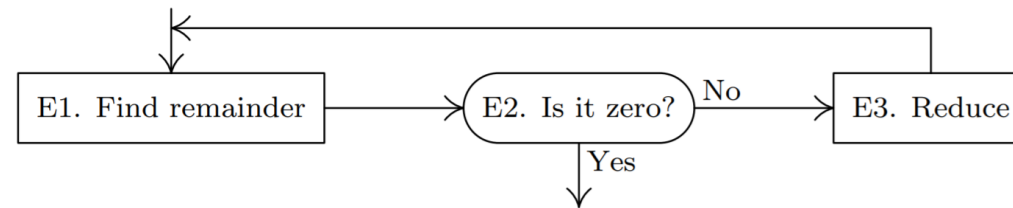


Fig. 1. Flow chart for Algorithm E.



Algoritmo e programma

- La rappresentazione dell'algoritmo comprensibile ed eseguibile dall'esecutore automatico costituisce un **programma**
- L'elaborazione delle azioni è richiesta all'elaboratore tramite comandi elementari chiamati **istruzioni** espresse attraverso un opportuno formalismo: il **linguaggio di programmazione**



Algoritmo e programma

- Un programma è un testo scritto in accordo al lessico, alla sintassi e alla semantica di un linguaggio di programmazione.
- Un programma è la formulazione testuale, in un certo linguaggio di programmazione, di un algoritmo che risolve un dato problema.

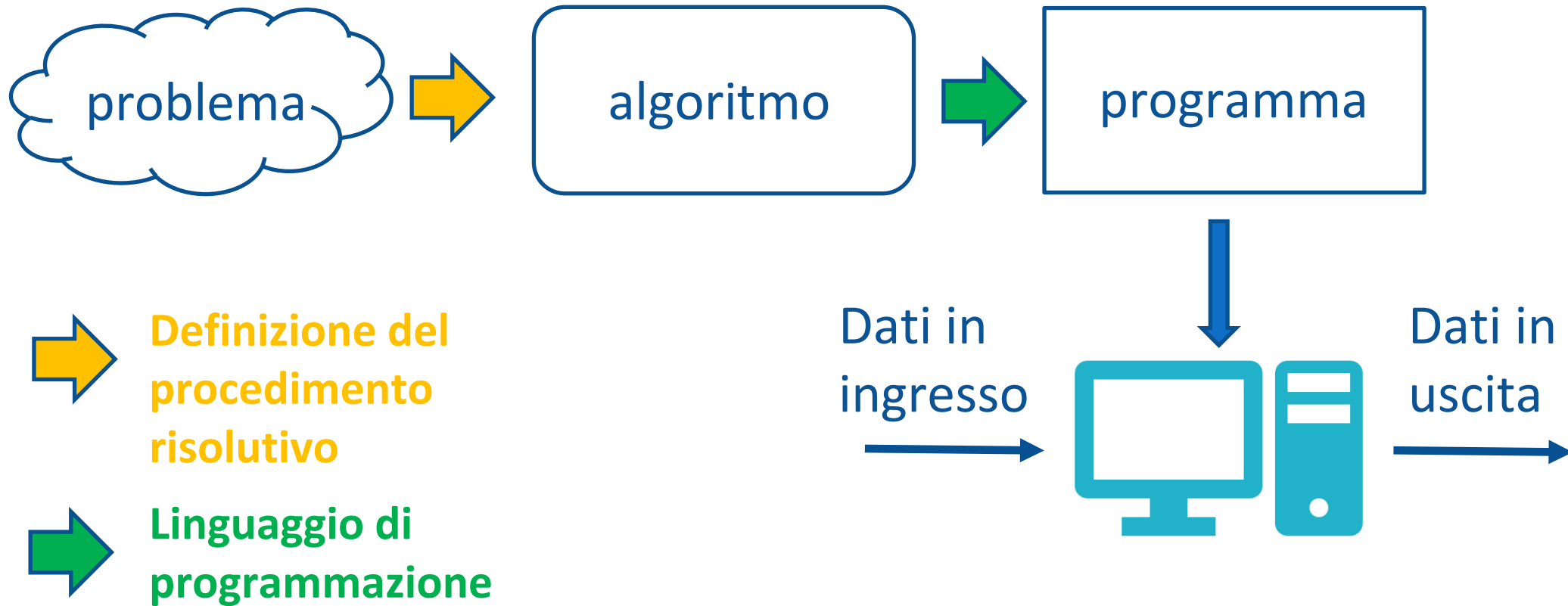


Algoritmo e programma

- Molti linguaggi di programmazione possibili, ognuno tipicamente indicato per risolvere problemi in un certo ambito
 - C, C++, Java, Python, Ruby, Pascal, Perl, Swift, FORTRAN, COBOL, Assembly, C#, Basic, R, Matlab, ...
- Di conseguenza, un algoritmo può essere implementato in linguaggi diversi
- Ognuno dei programmi ottenuti è **un'implementazione dell'algoritmo originale** ed è quindi **equivalente agli altri programmi** (assumendo che non ci siano errori ...)



Algoritmo e programma



Eseguire un programma: risorse e processo

- Una volta ottenuto il **programma**, questo deve essere eseguito su un sistema di elaborazione che realizzi il ruolo di **esecutore**.
- A questo scopo è necessario che il sistema metta a disposizione del programma le opportune **risorse** che rendano possibile l'esecuzione
 - Esempio: memoria per ospitare i dati ed il programma, dispositivi per inserire i dati in ingresso e fornire i dati in uscita, ecc.



Eseguire un programma: risorse e processo

- Se le risorse sono disponibili, è possibile avviare l'esecuzione del programma. Da questo momento in poi si parla di **processo**.
- Differenza sostanziale: mentre il **programma** è la descrizione di un procedimento risolutivo, il **processo** è invece l'attuazione di tale procedimento.
- Il processo richiede un esecutore che fornisce le risorse necessarie, effettua le azioni previste, riceve i dati di ingresso e produce i dati in uscita.



Eseguire un programma: risorse e processo

- Quali azioni compie l'esecutore per eseguire un programma? Possiamo descriverle *con l'aiuto di un algoritmo*

1. Fase di inizializzazione
 - a. verifica che tutte le risorse necessarie siano disponibili
 - b. inizializza un segnaposto alla prima istruzione del programma
2. Esegui ripetutamente i seguenti passi, fino al raggiungimento della fine del programma:
 - a. leggi l'istruzione del programma associata al segnaposto, e sposta in avanti il segnaposto
 - b. se l'istruzione ha bisogno di dati, acquisiscili
 - c. esegui l'istruzione

