

Informatica per il Turismo

Lezione 1:

Nozioni preliminari

INFORMAZIONE E COMUNICAZIONE

◆ Informazione

- Notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere, diminuendo l'incertezza

◆ Messaggio

- Tutto ciò che (tras)porta informazione

◆ Comunicazione

- Scambio di messaggi

INFORMATICA

- ◆ Scienza della rappresentazione e dell'elaborazione dell'informazione
- ◆ L'informatica studia le caratteristiche dell'informazione e i modi di usarla, immagazzinarla, trasportarla e manipolarla in modo automatico
- ◆ L'informatica ha due anime:
 - tecnologica: i calcolatori elettronici e i sistemi che li utilizzano
 - metodologica: i metodi per la soluzione di problemi e la gestione delle informazioni

ELABORATORE ELETTRONICO

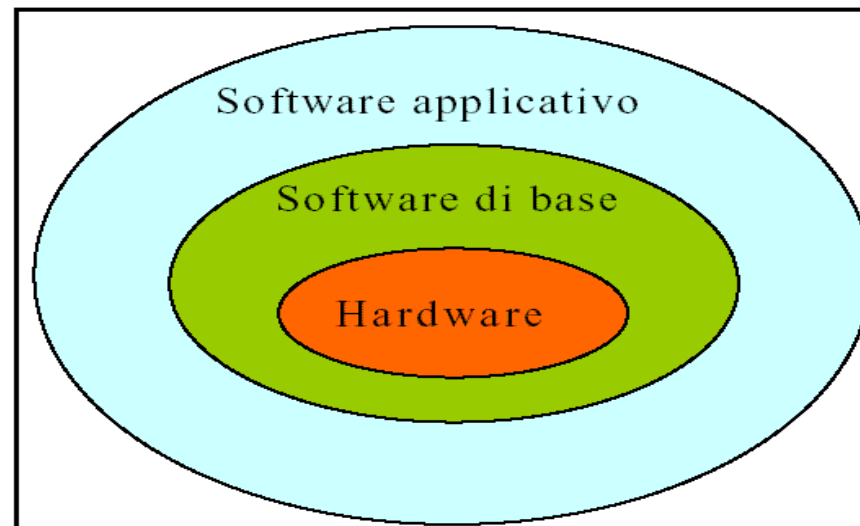
- ◆ Elaboratore elettronico (o *computer* o *calcolatore*)
 - è una macchina per la rappresentazione, la memorizzazione e l'elaborazione e la trasmissione (reti di calcolatori) delle informazioni
- ◆ Le principali componenti di un calcolatore si distinguono in componenti:
 - **Hardware**
 - ✓ la struttura fisica del calcolatore, costituita da componenti elettronici ed elettromeccanici
 - **Software**
 - ✓ l'insieme dei programmi che consentono all'hardware di svolgere dei compiti utili

HARDWARE E SOFTWARE

- ◆ L'hardware è la macchina reale
 - Le operazioni (chiamate istruzioni) che l'hardware sa eseguire direttamente rappresentano le frasi del linguaggio macchina del calcolatore
 - Le istruzioni del linguaggio macchina sono molto semplici e il calcolatore può eseguirle in modo molto efficiente
- ◆ Il software ha lo scopo di mostrare agli utenti il calcolatore come una macchina virtuale (non esistente fisicamente) più semplice da usare rispetto all'hardware sottostante

CLASSIFICAZIONE DEL SOFTWARE

- ◆ Software di Base (ad esempio il Sistema Operativo)
 - permette una più semplice interazione con le componenti hardware (memorie, periferiche, ...)
- ◆ Software Applicativo
 - consente agli utenti di utilizzare il calcolatore come una macchina virtuale utile per la **risoluzione di problemi**



PROBLEMI

◆ Problemi di varia natura, semplici e complessi

- Trovare il maggiore fra due numeri
- Dato un elenco di nomi e numeri di telefono, trovare il numero di una data persona
- Dati a e b , risolvere l'equazione $ax+b=0$
- Stabilire se una parola precede alfabeticamente un'altra
- Ordinare un elenco di nomi
- Creare, modificare e alterare suoni
- Analizzare, riconoscere e modificare immagini
- Gestire la contabilità di un'azienda
- Supportare operazioni di commercio elettronico

PROBLEMI

◆ Descrizione del problema

- La descrizione del problema non indica direttamente (in genere) un modo per risolverlo; in altre parole, esiste una differenza sostanziale tra la **descrizione** o **specifica** di un problema e specifica del **processo di risoluzione**

◆ Risoluzione di un problema

- Comprensione
- Modellazione
- Individuazione di un opportuno metodo risolutivo (**algoritmo di risoluzione**)

PROBLEMI

L'obiettivo fondamentale

Descrizione di un problema



Individuazione di un ALGORITMO

ALGORITMO

◆Etimologia

- La parola **algoritmo** deriva dal nome di un autore scientifico persiano del IX secolo, **Abu Ja'far Mohammed ibn Musa al-Khowarizmi**, il quale scrisse, circa nell'825, il trattato "Kitab al jabr w'al-muqabala" (forse *regole di trasposto e semplificazione*) dove descrisse delle regole per la semplificazione delle equazioni (il termine originario, **agorismo**, trasformato in **algoritmo** per analogia con **aritmetica**).

◆Significato moderno

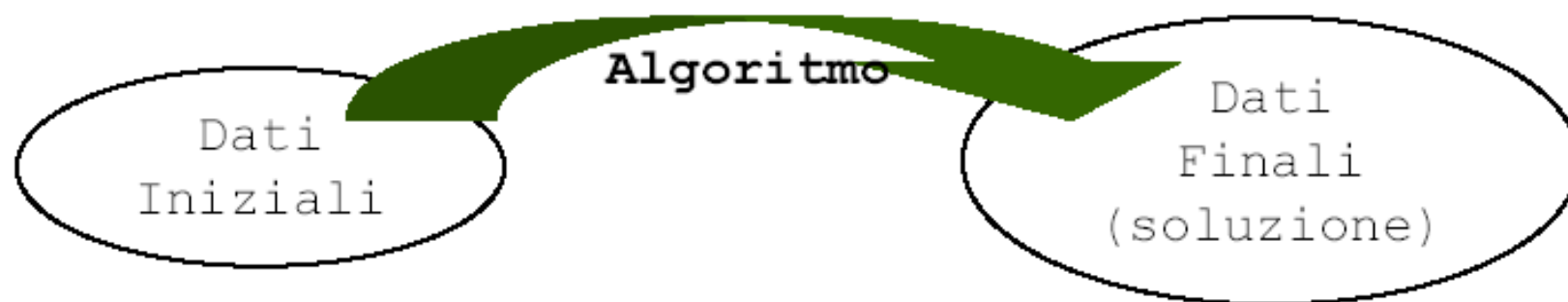
- Sequenza finita di passi che risolve in un tempo finito un problema.

ALGORITMO

◆ Algoritmi di varia natura, “semplici” e “complessi”

- Preparazione di un ricetta
- Istruzioni di montaggio di un armadio
- Prelievo bancomat
- Calcolo del massimo comun divisore tra due interi
- Istruzioni per conseguire la laurea ... con il massimo dei voti e con il minor sforzo possibile (algoritmo ottimo!!!)

ALGORITMO

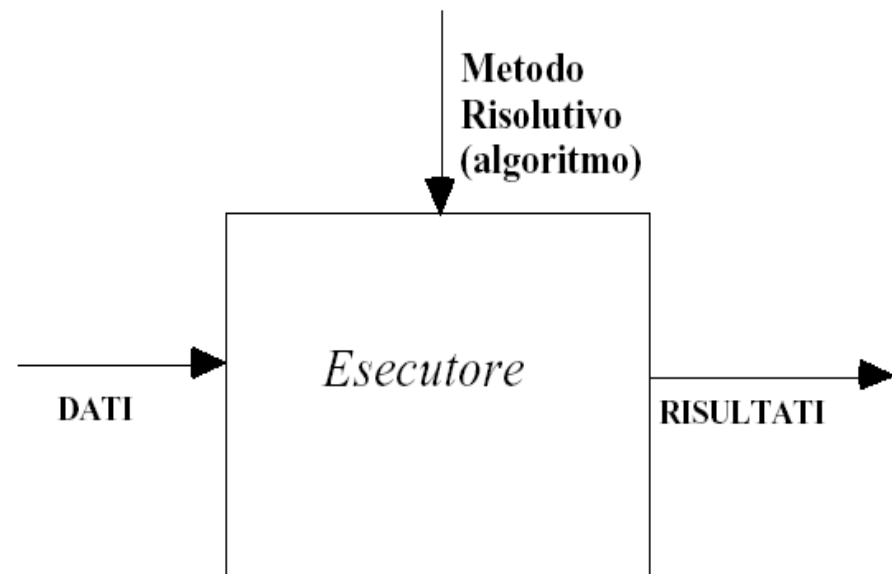


Si definisce *algoritmo* una *sequenza di azioni* che trasformi i dati iniziali in un numero finito di passi, elementari e non ambigui, per giungere al risultato finale.

Questa sequenza di azioni è valida per un insieme di dati iniziali ben definito e può essere eseguita da un opportuno esecutore.

ESECUTORE DI UN ALGORITMO

Esecutore: una *macchina astratta* capace di eseguire le azioni specificate dall'algoritmo.



PROPRIETÀ DEGLI ALGORITMI

◆ Proprietà fondamentali

- **Non-ambiguità**: ogni azione deve essere **univocamente interpretabile** dall'esecutore
- **Eseguibilità**: ogni azione deve essere **eseguibile in un tempo finito** da parte dell'esecutore dell'algoritmo
- **Finitezza**: per ogni insieme di dati di ingresso, il numero totale di azioni da eseguire deve essere **finito**

◆ Proprietà desiderabile

- **Efficienza**: deve risolvere il problema utilizzando al meglio le risorse a disposizione

CLASSIFICAZIONE DEI PROBLEMI E COMPLESSITA' DEGLI ALGORITMI

◆ Problemi

- Trattabili

- esistono algoritmi con costo polinomiale

- Intrattabili

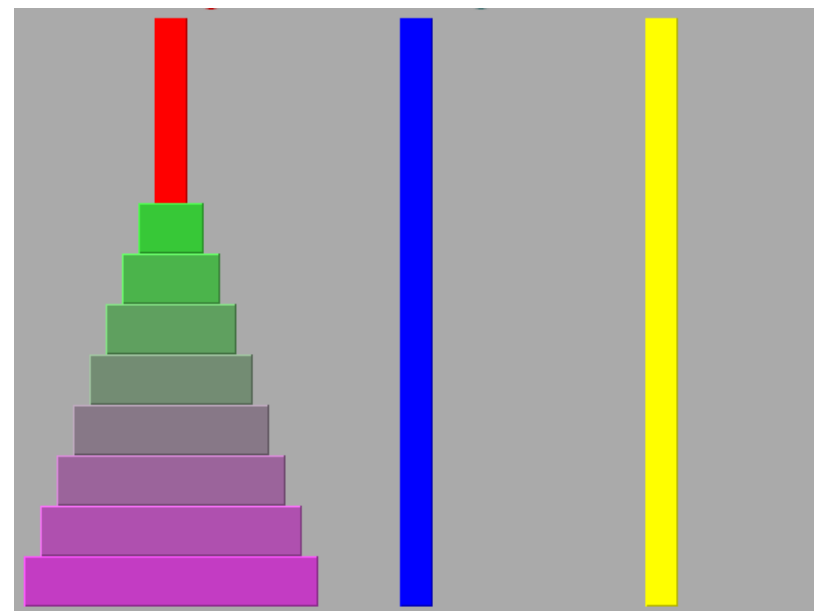
- non esistono algoritmi con costo polinomiale

- Non resolvibili

- non esistono algoritmi

UN PROBLEMA INTRATTABILE: LE TORRI DI HANOI

- $n = 64$ (dischi)
- mosse = $2^n - 1 = 2^{64} - 1 = 18'446'744'073'709'551'615$
- 1 mossa al giorno = $5,0539025 \cdot 10^{16}$ anni
- età della Terra = 4'600'000'000 anni
- ci restano 50 milioni di miliardi di anni !



RAPPRESENTAZIONE DEGLI ALGORITMI

- | | |
|---------------------------------|------------------------|
| 1. Linguaggio naturale | Linguaggi informali |
| 2. Diagrammi di flusso | Linguaggi semi-formali |
| 3. Pseudo-codice | |
| 4. Linguaggio di programmazione | Linguaggi formali |

ESEMPIO DI ALGORITMO: CALCOLO DEL MCD

Problema:

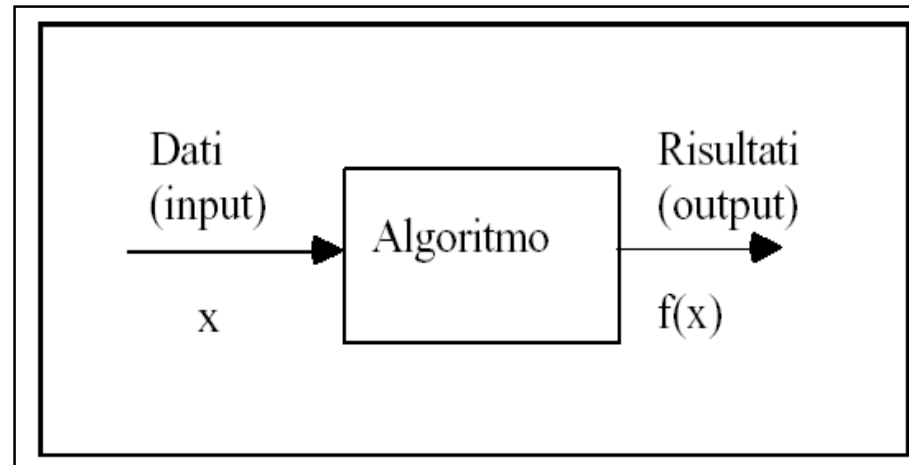
Calcolo del Massimo Comun Divisore (MCD) fra due interi M ed N

Algoritmo n° 1

1. Calcola l'insieme A dei divisori di M
2. Calcola l'insieme B dei divisori di N
3. Calcola l'insieme C dei divisori comuni di M e N, $C = A \cap B$
4. Il risultato è il massimo dell'insieme C

ALGORITMI EQUIVALENTI

- ◆ In generale un algoritmo può essere visto come una **funzione** da un dominio di ingresso (**input**) ad dominio di uscita (**output**)



- ◆ Due algoritmi si dicono **equivalenti** quando:
 - hanno stesso dominio di ingresso e stesso dominio di uscita
 - in corrispondenza degli stessi valori nel dominio di ingresso producono gli stessi valori nel dominio di uscita.

ESEMPIO DI ALGORITMO: CALCOLO DEL MCD

Algoritmo n° 2

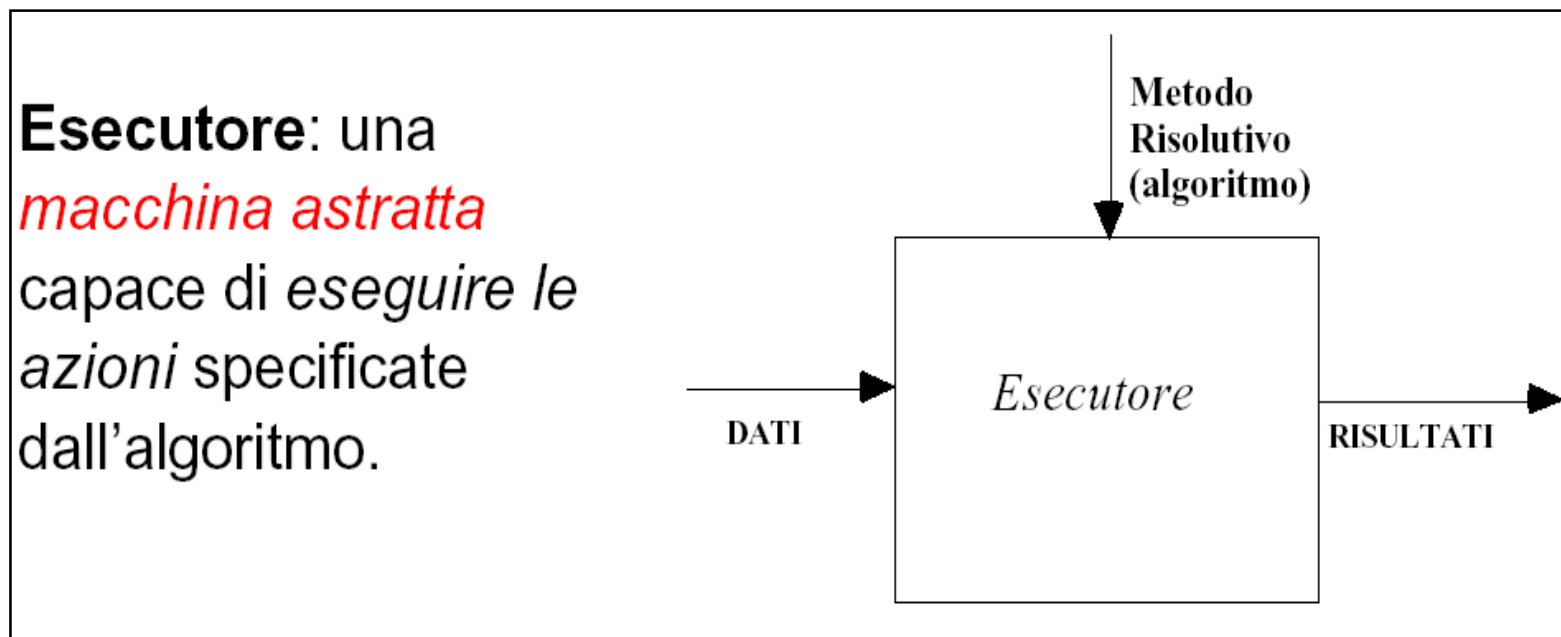
$$\text{MCD}(M,N) = \begin{cases} M \text{ (oppure } N) & \text{se } M=N \\ \text{MCD}(M-N, N) & \text{se } M>N \\ \text{MCD}(M, N-M) & \text{se } M<N \end{cases}$$

Strategia risolutiva:

- Finché $M \neq N$:
 - se $M > N$, sostituisci a M il valore $M' = M - N$
 - altrimenti sostituisci a N il valore $N' = N - M$
- Il Massimo Comun Divisore è il valore finale ottenuto quando M e N diventano uguali

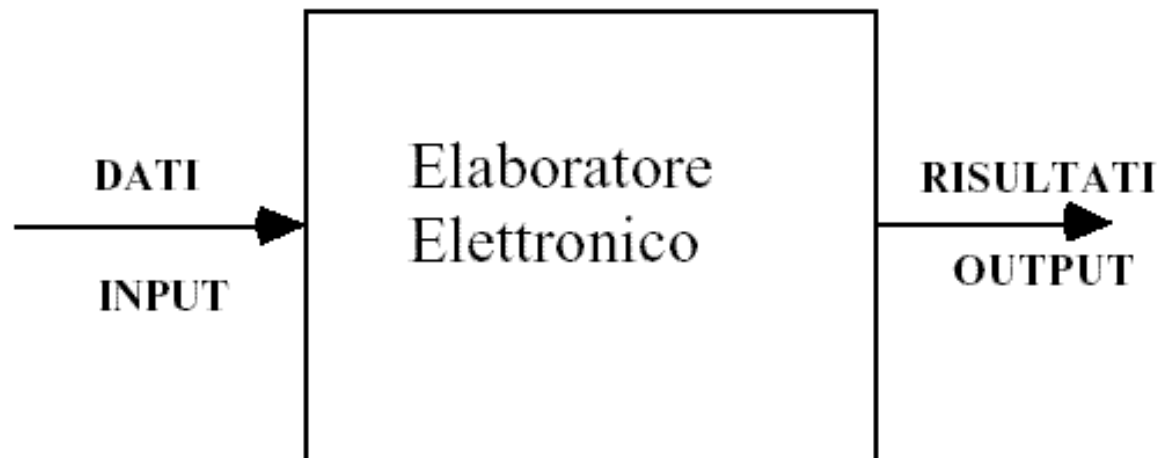
ALGORITMI: FLUSSO DI ESECUZIONE

- L'**esecuzione** delle azioni nell'**ordine specificato** dall'algoritmo consente di ottenere, a partire dai dati di ingresso, i risultati che risolvono la particolare istanza del problema.



ALGORITMI: ESECUZIONE TRAMITE COMPUTER

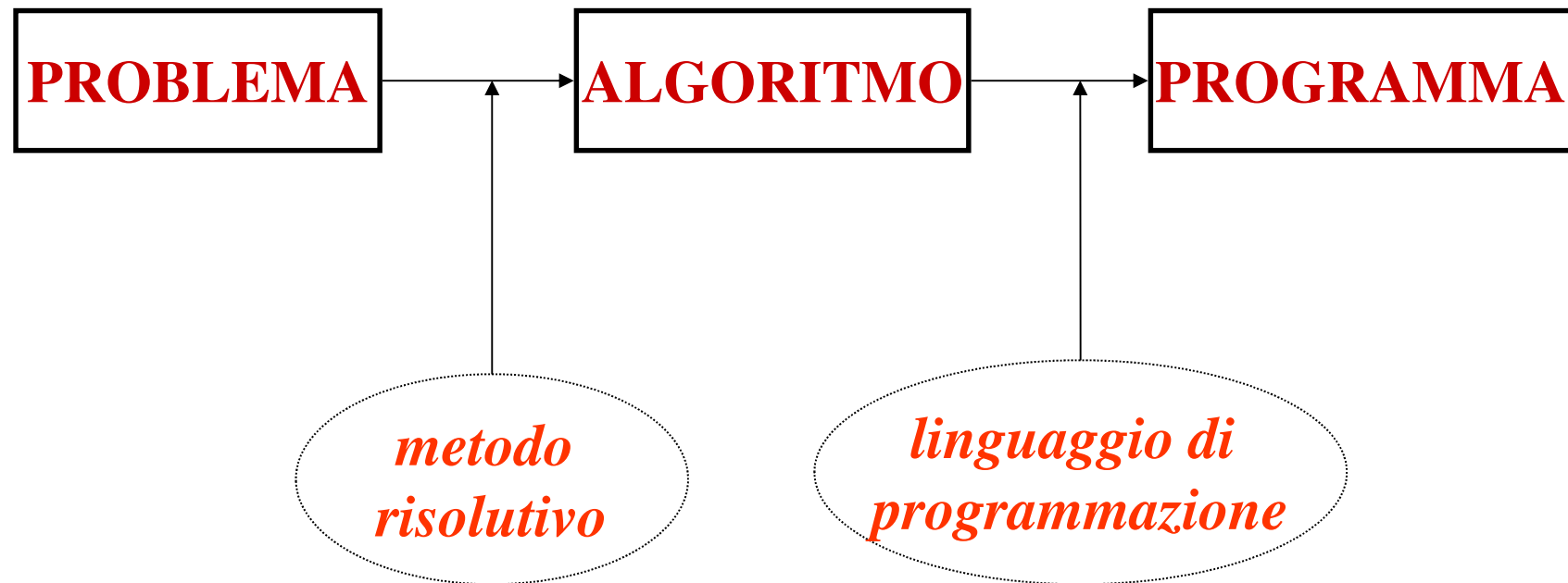
- ◆ La **programmazione** è l'attività con cui si predispone l'elaboratore ad eseguire un particolare **insieme di azioni** su una particolare tipologia di **dati**, allo scopo di risolvere un **problema**.



PROGRAMMA

- Ogni elaboratore è una macchina in grado di eseguire **azioni** elementari su **dati**
- L'esecuzione delle azioni elementari è richiesta all'elaboratore tramite comandi chiamati **istruzioni**
- Le istruzioni sono espresse attraverso **frasi** di un opportuno **linguaggio di programmazione**
- Un **programma** non è altro che la formulazione testuale di un algoritmo in un linguaggio di programmazione

ALGORITMI E PROGRAMMI



LINGUAGGI AD ALTO LIVELLO

È opportuno impostare la soluzione di un problema a partire dalle “mosse elementari” del **linguaggio macchina**?

- **si**, per risolvere il problema con **efficienza**
- **no**, se la macchina di partenza ha mosse di livello **troppo basso** (difficile progettare un algoritmo)



Linguaggi di Programmazione ad Alto Livello (di astrazione)

- le istruzioni corrispondono ad operazioni più complesse
- esempi: Pascal, Basic, C, C++, Java

Astrazione: processo di aggregazione di informazioni e dati e di sintesi di modelli concettuali che ne enucleano le proprietà rilevanti escludendo i dettagli inessenziali