

Informatica

Risolvere Problemi

Francesco Calimeri

Dipartimento Di Matematica

Università della Calabria

calimeri@mat.unical.it

<http://www.mat.unical.it/calimeri>

Grazie a Francesco Ricca

<http://www.mat.unical.it/~ricca>

Cosa possiamo fare con un calcolatore?

- Word Processing. Memorizzare, elaborare testi.
- Basi di Dati. Memorizzare grossi archivi di dati, recupero veloce, produrre informazioni globali.
- Accesso Remoto. Trasmissione e recupero di informazioni.
- Calcolo. Risolvere problemi matematici.
- Simulazioni. Rappresentare e elaborare informazioni che simulano l'ambiente reale.

Parole chiave:

- Hardware
- Software

Problemi, Algoritmi e Programmi

I concetti fondamentali dell'Informatica



Il problema

- Abbiamo un problema quando ci poniamo un obiettivo da raggiungere e per raggiungerlo dobbiamo mettere a punto una strategia



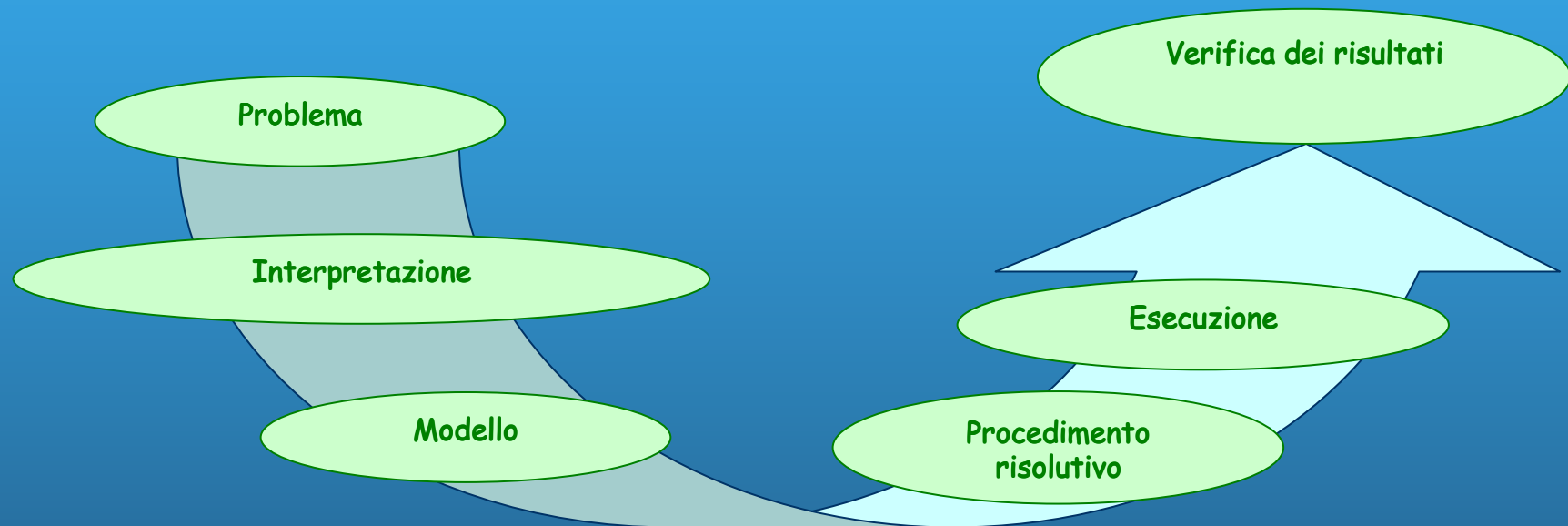
Esempi di problemi

- Trovare il numero di telefono di una persona
- Individuare il numero più piccolo di una sequenza
- Stabilire se una parola precede alfabeticamente un'altra
- Calcolare il costo totale di un certo numero di prodotti
- Trovare perimetro e area di una figura geometria
- ...

Risolvere un problema

- Come si costruisce la soluzione a un problema?
- Qual è il giusto “punto di partenza” per pensare la soluzione a un problema?
- Quali metodologie e tecniche usare?

Risolvere un problema



Algoritmo

- Un algoritmo è una **sequenza finita di operazioni** elementari che porta alla **risoluzione in un tempo finito** una classe di problemi.
- In generale un algoritmo può essere visto come una funzione da un dominio d'ingresso ad uno d'uscita



Algoritmi: proprietà fondamentali

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

Algoritmi equivalenti

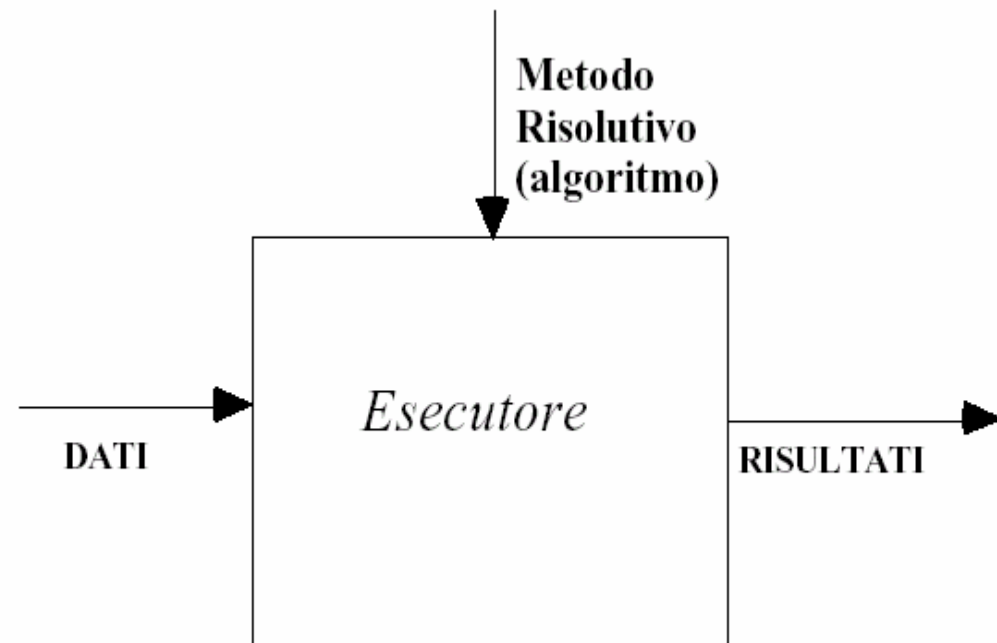
- forniscono lo stesso risultato
- ma possono avere diversa efficienza
- e possono essere profondamente diversi !
- Esempio: moltiplicare tra loro due numeri
- **Algoritmo 1:** Somme successive:
 - $12 \times 12 = 12 + 12 + \dots + 12 = 144$

Algoritmo 2

12x
12=
24
12=
144

Esecuzione

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



Algoritmi e programmi

- **Algoritmo**

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

- **Codifica**

- Fase di scrittura di un algoritmo in un qualche **linguaggio di programmazione**, che specificano le azioni da compiere.

- **Programma**

- Testo scritto in accordo con la sintassi e la semantica di un linguaggio di programmazione.



Linguaggi di Programmazione

- Linguaggi per esprimere in maniera rigorosa un algoritmo
- Linguaggio macchina (seq. Istruzioni)
- Linguaggi ad alto livello
(*vicini al ling. naturale*)
 - Pascal, C e C++, Basic, Cobol, Java

Esempio ...in Java

```
int potenza(int a, int n)
{
    int ris = 1;
    while ( n > 0 ) {
        ris = ris * a;  n = n - 1;
    }
    return ris;
}
```

Riassumendo...

- Un **algoritmo** è il **processo risolutivo** di un problema
- Ogni elaboratore è una macchina in grado di **eseguire azioni elementari** su dati **dette istruzioni**
- Le istruzioni sono espresse attraverso frasi di un opportuno **linguaggio di programmazione**
- Un **programma** è la formulazione testuale di un **algoritmo** in un **linguaggio di programmazione**

Esistono problemi che un elaboratore non può risolvere?

- **Sì.** Ci sono problemi non calcolabili da nessun modello di calcolo reale o astratto
- Esempio: data una funzione $f : \mathbb{N} \rightarrow \mathbb{N}$, **stabilire se $f(x)$ è costante** per ogni valore di x

Linguaggi di Programmazione

- Linguaggi per esprimere in maniera rigorosa un algoritmo
- La “lingua del Computer”
- Diversi tipi di linguaggi:
 - Imperativi
 - Ad Oggetti
 - Dichiarativi

Linguaggi Imperativi

- “Comandano” il computer indicando le azioni da compiere
- Si specificano **le azioni da fare** per ottenere il risultato
- Algoritmo = “insieme di operazioni da compiere sui dati”

Algoritmo = DATI + Controllo

Linguaggi ad Oggetti

- Sono basati sul concetto di **oggetto software**
- Un oggetto software **rappresenta un oggetto del mondo reale**
 - un numero, un archivio, un testo, una matrice, una persona...
- I dati sono rappresentati come oggetti
- Le azioni da compiere come operazioni da effettuare sugli oggetti.
- Di solito sono realizzati come estensione dei linguaggi imperativi.
- **Un programma modella un problema reale come una collezione di oggetti software che interagiscono.**

Linguaggi Dichiarativi

- “Chiedono” al computer di trovare una soluzione
- Si specifica **quello che si vuole** e non come deve essere calcolato
- Linguaggi Funzionali: Lisp
- Linguaggi Logici: Prolog, ASP/DLP
- Questo è stato espresso da Robert Kowalski con la formula:
Algoritmo = Logica + Controllo

Ma è davvero questa la lingua del computer?

- Questi sono linguaggi ad *alto livello*
 - Più simili al nostro modo di esprimerci
- Il computer “conosce” solo i bit e le operazioni elementari su questi (*linguaggio a basso livello*)
- C'è bisogno di un traduttore!!!
- Il **Compilatore** traduce un programma scritto in un linguaggio di programmazione ad alto livello in bit e operazioni elementari
- Si trovano negli ambienti di sviluppo: e.s. JDK e/o Eclipse

Dal programma all'esecuzione...

