

Come “ragiona” il computer

Problemi e algoritmi



Il problema

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



Alcuni problemi tipici dell'informatica



Ricerca di informazioni

- Trovare il numero di telefono di una persona in un elenco
 - Individuare il numero più piccolo di una sequenza
 - Stabilire se una parola precede alfabeticamente un'altra
-



Problemi di elaborazione di informazioni

- Calcolare il costo totale di un certo numero di prodotti
- Trovare perimetro e area di una figura geometrica
- ...



Problemi di ottimizzazione

- Trovare tra tutte le soluzioni possibili del problema quella che rende minimo un certo fattore, per esempio scegliere il mezzo di trasporto più economico per andare a Parigi oppure quello con il quale si impiega meno tempo
-

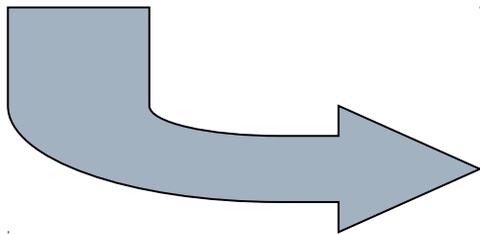


Risolvere un problema

- ❑ 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?
-

Descrizione procedimento risolutivo

- Individuazione di una sequenza di passi che, partendo dai dati noti, arrivi a dare la soluzione.



Definizione
Algoritmo



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

Due algoritmi si dicono **equivalenti** quando:

- hanno lo stesso dominio di ingresso;
 - hanno lo stesso dominio di uscita;
 - in corrispondenza degli stessi valori nel dominio di ingresso producono gli stessi valori nel dominio di uscita.
-

Algoritmi equivalenti

Due 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:

$$12 \times 12 = 12 + 12 + \dots + 12 = 144$$

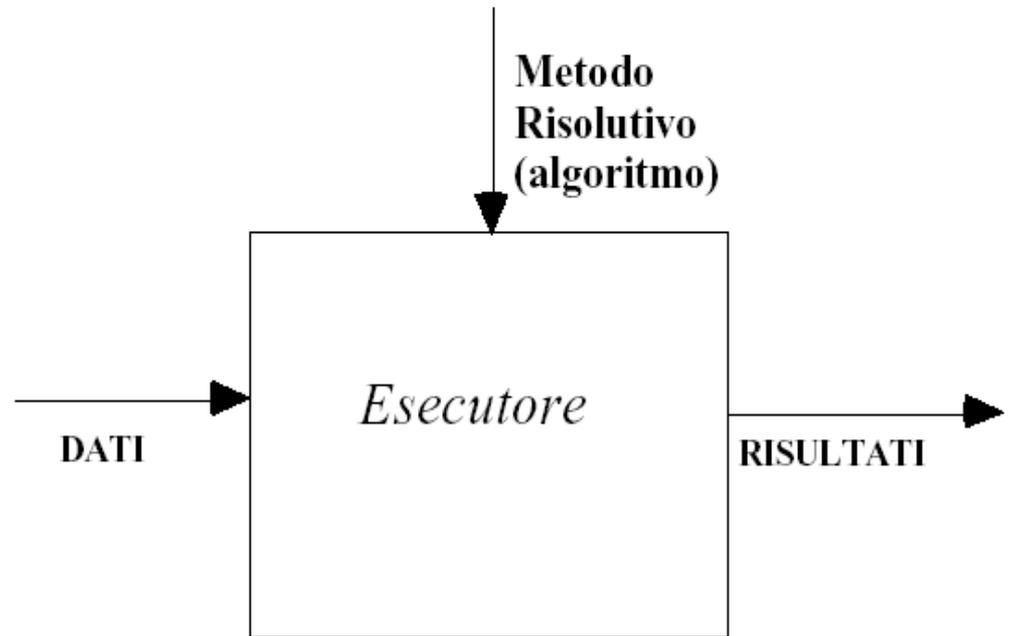
Algoritmo 2

“somma e shift”:

$$\begin{array}{r} \underline{12} \times \\ 12 \\ 24 \\ \hline 12 \\ \hline 144 \end{array}$$

Esecuzione

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



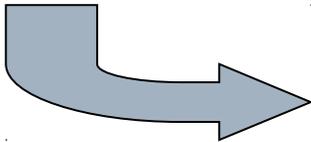
Algoritmi e programmi

□ Algoritmo

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

□ Codifica

Fase di scrittura di un algoritmo attraverso un insieme ordinato di frasi (“istruzioni”), scritte 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)
 - Esempi:
 - Pascal
 - C e C++
 - Java
 - Basic
-

Esempio di programma

Sub SOMMA()

Dim A, B as Integer

A = InputBox("Immetti un numero")

B = InputBox("Immetti un secondo numero")

Print "Somma: "; A+B

End Sub

Esempio: potenza

- **Problema:** Calcolare a elevato alla n (a^n)
 - Utilizziamo le **variabili** N , Ris
 - Inizialmente $Ris=1$ e $N=n$
 - **Algoritmo:**
 - Fino a che $N>0$
 - Calcola $Ris \times a$ e memorizzalo in Ris
 - Decrementa N
 - **Correttezza:**
 - Al termine $Ris=a^n$
-

Esempio in Pseudo Pascal

Program potenza;

Integer Ris,N,A;

Read(N);**Read**(A);

Ris=1;

While (N>0) **do**

 Ris=Ris*A;

 N=N-1;

Print(Ris);

Riassumendo...

- 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** è la formulazione testuale di un algoritmo in un linguaggio di programmazione
 - Un **algoritmo** è il processo risolutivo di un problema
-

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
-

Esistono problemi che un elaboratore non può risolvere?

- **Esempio.** Dato un insieme di immagini di paesaggi, determinare quello più rilassante.
 - Più in generale, quando il problema presenta **infinite** soluzioni, o non è stato trovato per esso un metodo risolutivo o è dimostrato che non esiste un metodo risolutivo
-

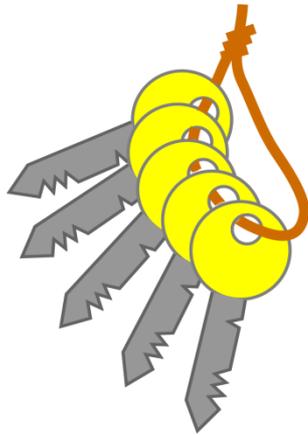
Diagramma di flusso o diagrammi a blocchi

- È uno dei metodi più comuni usati per la rappresentazione di algoritmi.
- Si presenta come un insieme di figure geometriche collegate da frecce.



Problema della chiave

- Trovare in un mazzo di chiavi quella che apre il lucchetto



Assunzioni:

- una tra le chiavi apre la porta
 - al buio, si prende una chiave a caso per volta
-



Inizio



- Tutti i diagrammi a blocchi cominciano con un'ellisse che contiene la parola **inizio**



Dati in ingresso

**Dati in
ingresso**

- I **dati in ingresso** sono i dati noti del problema, quelli che devono essere elaborati per arrivare alla soluzione



Operazioni

Operazioni

- Le **operazioni** da svolgere sui dati sono racchiuse in rettangoli



Scelta

- Quando si deve fare una **scelta** tra due possibilità si usa il rombo



Dati in uscita

**Dati in
uscita**

- I **dati in uscita** sono quelli che si vuole conoscere e costituiscono il risultato dell'elaborazione



Fine

- Ogni diagramma di flusso si conclude con un'ellisse che contiene la parola **fine**



Problema della stazione

□ Come si arriva alla stazione?



Operazioni elementari possibili:

- Andare avanti fino a un punto di incrocio
- Girare

