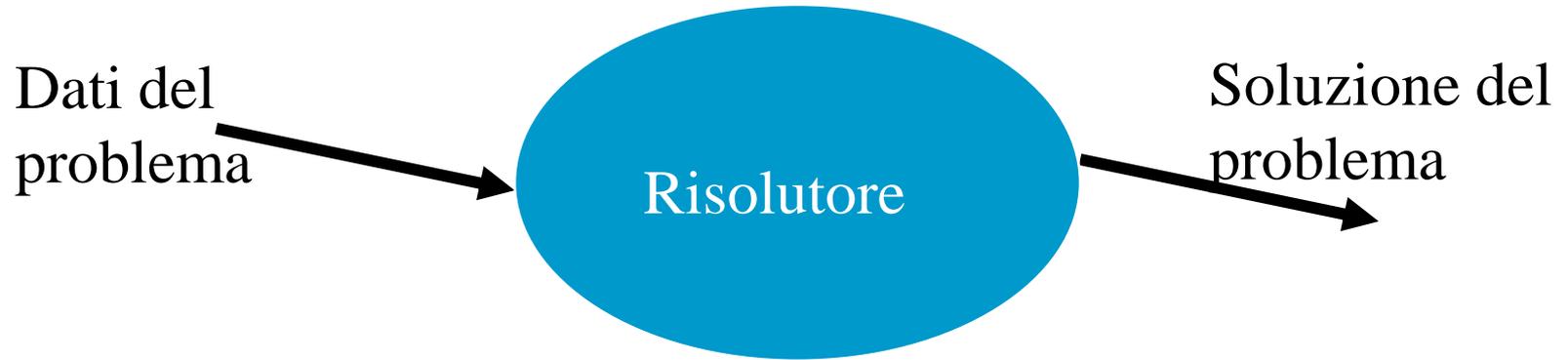


Cos'è l'informatica

# Cos'è l'informatica

- Scienza dei calcolatori elettronici (computer science)
- Scienza dell'informazione (information science)
  
- E' una scienza
- E' una tecnologia
  
- Cosa può essere automatizzato nell'elaborazione dell'informazione?
- Come lo si può automatizzare in modo efficace ed efficiente?



Metodi per la rappresentazione dell'informazione

Metodi per la descrizione delle modalità con cui ottenere una soluzione

Caratterizzazione del risolutore dei problemi

## **Problema**

Risolvere un problema significa ricercare ed esprimere un elenco di istruzioni che interpretate da un esecutore conducono da determinate informazioni iniziali ad altre informazioni finali soddisfacenti un criterio di verifica

Un problema è ben formulato se:

- non è a priori evidente che non esistono soluzioni
- il criterio di verifica delle soluzioni è univoco e si sa come applicarlo
- l'insieme dei dati iniziali è completo

Si definisce effettiva per un esecutore la soluzione di un problema allorchè:

- l'esecutore è in grado di interpretare i dati di ingresso
- l'esecutore è in grado di interpretare la descrizione di tale soluzione, e quindi di associare ad essa le azioni che deve compiere per eseguirla
- l'esecutore è in grado di compiere tali azioni, completando l'esecuzione in un tempo finito

# Algoritmo

- Un algoritmo è il procedimento risolutivo di un problema
- E' l'insieme di regole che, eseguite ordinatamente, permettono di ottenere i risultati del problema a partire dai dati a disposizione

Non tutte le regole formano un algoritmo

# Algoritmo

- Def. ISO: Un algoritmo è un insieme finito e ordinato di regole formalizzate destinato a fornire la soluzione di un problema, in un numero finito di passi
- Affinchè un insieme di regole possano essere considerare un algoritmo devono rispettare alcune proprietà

# Proprietà di un algoritmo

- **Non ambiguità:** ogni azione elementare indicata nell'algoritmo deve produrre un risultato ben definito, per tutti i possibili valori di ingresso
- **Eseguibilità:** ogni passo dell'algoritmo deve essere effettivamente eseguibile, in un tempo finito, dall'esecutore considerato
- **Determinismo:** nell'esecuzione dell'algoritmo, per ogni passo compiuto deve essere determinato uno e un solo passo successivo.

# Proprietà di un algoritmo

- **Finitezza:** l'algoritmo deve essere costruito da passi discreti e la sua lunghezza deve essere finita
- **Terminazione:** l'esecuzione dell'algoritmo deve, prima o poi, terminare.

# Esempi di algoritmi

- Calcolare la soluzione di  $ax + b = 0$ 
  1. leggere i valori di  $a$  e di  $b$
  2. calcolare  $-b$
  3. dividere  $-b$  per  $a$  e assegnare il risultato ad  $x$
  4. scrivere  $x$

# Esempi di algoritmi

- Problema del trasporto della capra, del lupo e del cavolo da una sponda ad un'altra del fiume
  1. Porta la capra sull'altra sponda
  2. Torna indietro
  3. Porta il cavolo sull'altra sponda
  4. Porta la capra indietro
  5. Porta il lupo sull'altra sponda
  6. Torna indietro
  7. Porta la capra sull'altra sponda

# Processo di risoluzione

- Analisi del problema ed identificazione di una soluzione
- Formalizzazione della soluzione e definizione dell'algoritmo risolutivo
- Scrittura di un programma in un linguaggio di programmazione
- Traduzione del programma in un linguaggio compreso dal calcolatore
- Esecuzione del programma

# Algoritmo

È un elenco finito di istruzioni univocamente interpretabili ciascuna delle quali deve essere precisamente definita e la cui esecuzione si arresta per fornire i risultati di una classe di problemi per ogni valore dei dati di ingresso

1. Dividere il primo numero per il secondo e chiamare  $r$  il resto
2. Se  $r=0$  il MCD è il più piccolo fra i due numeri
3. Se  $r$  è diverso da zero si prendono il secondo numero come primo numero e il resto come secondo numero
4. Si ricominci il procedimento