

## Introduzione

Sequenza finita di passi interpretabili da un esecutore.

Non è detto che l'esecuzione richieda un tempo finito.

## Tecnologie

Insertsort e mergesort: algoritmi di ordinamento.

- $(C_1 \cdot n \cdot n) = C_1$ : costante;  $n$ : numero di oggetti da ordinare.

- $(C_2 \cdot n \cdot \log_2 n)$

$C_1 = 2, C_2 = 50$

Stima tempo di esecuzione:  $n_i/n_s$

$n_i$ : numero di istruzioni

L'algoritmo mergesort risulta più efficiente, anche su un computer con minori prestazioni.

## classificazione

- Algoritmi sequenziali: eseguono un solo passo alla volta.
- Algoritmi paralleli: eseguono più passi alla volta, necessitano di più esecutori.

In base al metodo di risoluzione delle scelte:

- Algoritmi deterministici: utilizza dei criteri precisi es risultato di un'espressione)
- Algoritmi probabilistici: effettua una scelta basata sulla casualità.
- Algoritmi non deterministici: categoria teorica, segue in ipotesi tutte le possibili alternative, supponendo di avere risorse computazionali illimitate.

## Problemi e correttezza

$I$  = insieme dati di input  
 $S$  = insieme soluzioni

$$P \subseteq I \times S$$

$P$  = Problema

Tipi di problemi:

- Decisione:  $S = \{0, 1\}$ , decide se una cosa è vera o meno.
- Ricerca: trovare una generica soluzione  
Per ogni istanza dei dati in ingresso.
- Ottimizzazione: trovare la soluzione ottima rispetto ad un criterio prefissato.

Problema  $\rightarrow$  cosa, algoritmo  $\rightarrow$  come.

Correttezza: dati  $P$  e un algoritmo, quest'ultimo termina e produce la soluzione corretta.

La correttezza si dice parziale se la terminazione non è sempre garantita.

## Complessità e risorse

Più algoritmi <sup>corretti</sup> possono essere confrontati in base a:

- Tempo di calcolo (complessità)
- Spazio di memoria
- Banda trasmissiva

Solitamente si considera per primo il tempo.