

# Introduzione all'Informatica e alla Programmazione

---

## Cosa è l'informatica?

---

Esistono diverse definizioni di informatica, quali:

- l'informatica è la scienza che si occupa della conservazione, dell'elaborazione e della rappresentazione dell'informazione.
- "Scienza dei calcolatori elettronici" (Computer Science)
- "Scienza dell'informazione"

Nel corso del nostro percorso di studio indichiamo l'Informatica come lo studio sistematico degli algoritmi che descrivono e trasformano l'informazione, la loro teoria, l'analisi e la progettazione degli algoritmi nonché la gestione dell'efficienza degli stessi con il fine di realizzare ed utilizzare strumenti per la risoluzione di problemi.

Come informazione intendiamo qualsiasi elemento che può essere rappresentato all'interno di un computer, quale:

- Numeri
- Caratteri e parole
- Immagini
- Suoni
- Filmati
- Comandi e sequenze di comandi che il calcolatore può eseguire

Lo strumento che si occupa di eseguire e gestire queste informazioni è il calcolatore elettronico, detto anche computer.

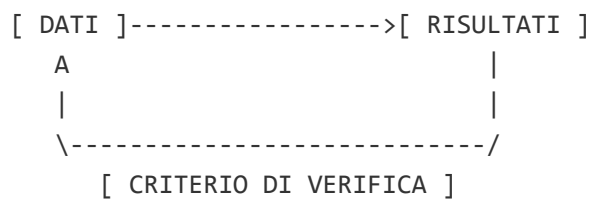
## Il significato di Programmazione

---

Possiamo indicare con programmazione l'attività con cui si predispone l'elaboratore ad eseguire un particolare insieme di azioni su particolari informazioni di origine, allo scopo di risolvere un certo problema.

Risolvere un problema significa ricercare e descrivere un processo risolutivo che, eseguito, consente, a partire da delle informazioni iniziali (dati), di ottenere delle informazioni finali

(risultati) soddisfacenti ad un criterio di verifica secondo lo schema descritto nella seguente figura:



Si afferma che un problema è ben formulato o formalmente definito se soddisfa le seguenti specifiche:

1. Il problema è formulato in una forma comprensibile dal solutore
2. I dati sui quali si deve cercare la soluzione sono completi e coerenti
3. È un ivoco il criterio di verifica della soluzione

## La formalizzazione dei problemi

Il processo di formalizzazione o modellazione di un problema consiste nel tradurre un problema reale in una sua formulazione alternativa, alquanto semplificata ma equivalente per quanto riguarda la tematica di interesse, in modo tale che esso possa essere risolto automaticamente. Applicando l'inverso si questa trasformazione si tradurranno poi immediatamente i risultati nell'ambito originale del problema.

## Trasformazione dei problemi

Non sempre l'ambito nel quale sorgono e vengono formulati i problemi è il contesto più idoneo nel quale ricercare e descrivere la soluzione. Risulta spesso comodo eseguire la trasformazione in un altro ambito nel quale risulta più facile risolvere il problema. In questo caso si parla quindi di passaggio dello spazio dei problemi allo spazio delle soluzioni.

## Classificazione dei Problemi

I problemi, come vedremo successivamente, sono classificabili secondo criteri ben definiti rispetto all'algoritmo che ne consente la risoluzione; un problema è *intrattabile* se la sua complessità computazionale lo rende di fatto non risolvibile, è *irrisolto* se non esiste o non si è ancora individuato un algoritmo di risoluzione.

I problemi però possono essere anche classificati in quattro grandi categorie:

- Problemi di *decisione* (recognition): quando la soluzione deve fornire come soluzione una scelta tra due possibili alternative, quindi la risposta sarà *vero* o *falso*.

- Problemi di *ottimizzazione* (optimization): quando si deve effettuare una scelta tra possibili alternative e deve fornire come soluzione la migliore tra le possibili.
- Problemi di *enumerazione*: quando si deve trovare un valore come risultato calcolato.
- Problemi di *ricerca*: quando è necessario individuare un valore esplorando in maniera sistematica lo spazio dei dati a disposizione.

## Gli algoritmi

---

L'algoritmo è il procedimento attraverso cui è possibile arrivare ad una soluzione di un problema attraverso una serie predefinita di operazioni codificate. Identificando con *Istanza del problema* la formulazione del problema quando le sue variabili assumono un valore particolare, possiamo indicare come *algoritmo risolutore* l'algoritmo A che risolve un problema P se applicato ad ogni sua istanza I ne produce la soluzione corretta.

La generalità dell'algoritmo, ci permette di affermare che la risoluzione del problema implica la risoluzione di tutte le sue istanze che quindi devono fornire il corretto risultato al variare di tutti i dati in ingresso. Da quanto si evince esiste una stretta relazione tra algoritmo e problema, pertanto se individuiamo dei criteri per la classificazione degli algoritmi sarà possibile classificare e definire la complessità dei problemi utilizzando i medesimi parametri che definiscono la qualità degli algoritmi. E' anche noto che è possibile identificare diversi algoritmi risolutivi per uno specifico problema, diversi sia per prestazioni che per qualità; sarà quindi necessario stabilire per ogni algoritmo un parametro specifico che ne identifica le caratteristiche e successivamente, considerando gli algoritmi che risolvono lo stesso problema, classificarli nella giusta classe di appartenenza.

## Parametri di qualità di un algoritmo

La scrittura di un algoritmo comporta scelte progettuali che portano all'identificazione di una strategia risolutiva, per definire la qualità di un algoritmo dobbiamo quindi tenere presente:

- la *semplicità* di comprensione e modificabilità del codice;
- la *qualità di scrittura* del codice del programma;
- la *velocità* di esecuzione dell'algoritmo;
- ...

per una classificazione univoca non è possibile utilizzare parametri soggettivi, è quindi necessario identificare dei parametri oggettivi in modo da ottenere un criterio di classificazione universale per la loro classificazione. Alcuni dei parametri oggettivi universalmente riconosciuti ed utilizzati sono:

- La complessità temporale: ovvero il tempo impiegato per l'esecuzione dell'algoritmo.
- La complessità spaziale: cioè la dimensione del programma, nonché dell'occupazione

di memoria necessaria all'esecuzione dell'algoritmo.

- La complessità di I/O: cioè il tempo impiegato per l'acquisizione e output delle informazioni necessarie alla risoluzione.
- La complessità di trasmissione: ovvero la capacità di trasferimento di informazioni tra entità remote.

## La complessità temporale

La complessità temporale di un algoritmo indica la quantità di tempo impiegata da un algoritmo a essere eseguito in funzione della lunghezza della stringa che rappresenta l'input; questa è espressa comunemente usando la notazione O-grande, che esclude i coefficienti e i termini di ordine inferiore. Quando è espressa in questo modo, si dice che la complessità temporale è descritta asintoticamente, ossia, quando la dimensione degli input va a infinito. Ad esempio, se il tempo richiesto da un algoritmo su tutti gli input di dimensione  $n$  è al massimo  $5n^3 + 3n$ , la complessità temporale asintotica è  $O(n^3)$ .

La complessità temporale è stimata comunemente contando il numero di operazioni elementari eseguite dall'algoritmo, dove un'operazione elementare impiega una quantità di tempo fissa a essere eseguita. Così la quantità di tempo impiegata e il numero di operazioni elementari eseguite dall'algoritmo differiscono al massimo di un fattore costante.

Poiché il tempo di esecuzione di un algoritmo può variare con diversi input della stessa dimensione, si usa comunemente la complessità temporale del caso peggiore di un algoritmo, denotata come  $T(n)$ , che è definita come la quantità di tempo massimo impiegata su qualsiasi input di dimensione  $n$ . Le complessità temporali sono classificate in base alla natura della funzione  $T(n)$ . Ad esempio, un algoritmo con  $T(n) = O(n)$  è chiamato algoritmo in tempo lineare, e un algoritmo con  $T(n) = O(2^n)$  si dice che è un algoritmo in tempo esponenziale.

Per ulteriori informazioni sulla notazione O-grande si rimanda a

<https://it.wikipedia.org/wiki/O-grande>

## Rappresentazione di un algoritmo

Gli algoritmi sono procedure che devono poter essere descritte in maniera formale e possano essere eseguite da strumenti automatici seguendo una serie di step predefiniti. Per questo motivo vi sono forme di rappresentazione grafica universalmente riconosciute che consentono di rendere comprensibile, potete trovare una breve informativa rispetto all'uso di diagrammi di flusso agli indirizzi:

- [https://it.wikipedia.org/wiki/Diagramma\\_di\\_flusso](https://it.wikipedia.org/wiki/Diagramma_di_flusso)
- <http://security.polito.it/~lioy/12bhd/programmare.pdf>

# Classificazione degli algoritmi

Gli algoritmi possono essere suddivisi in due tipologie:

- *deterministici*: se per ogni istruzione esiste, a partita di dati in ingresso, un solo passo successivo.
- *non deterministici*: se contiene almeno un istruzione che ammette più passi successivi.

Poiché gli algoritmi sono la soluzione di un problema, possiamo definire un problema  $P$  come *computabile* se esiste un algoritmo  $A$  che lo risolve, cioè per cui applicando l'algoritmo  $A$  ad una qualunque istanza  $I$  di problema  $P$  ne fornisce l'esatta soluzione.

Gli algoritmi possono poi essere suddivisi in classi in funzione della loro complessità in:

- Lineari
- Esponenziali
- Logaritmici
- Fattoriali
- ...

in maniera molto simile a quanto visto per la classificazione dei problemi secondo la teroa degli O-grandi.

## Esercizi:

---

Provare ad identificare quali dei problemi di seguito riportati è ben formulato e se lo è descrivere il procedimento risolutivo dello stesso:

1. Dato il perimetro e l'area di un triangolo, determinare le misure dei lati
2. Dato il perimetro e l'area di un rettangolo, determinare la misura del lato ad esso equivalente
3. Dato il perimetro e l'area di un rettangolo, determinare la misura della diagonale
4. Dato il perimetro di un rettangolo, determinarne l'area
5. Dato il perimetro e l'area di un rettangolo, detrminare la misura dei lati
6. Noti i cateti di un triangolo rettangolo calcolre l'altezza relativa alla ipotenusa