

Lezione 1

4 ottobre 1999

Argomenti trattati

- Presentazione del corso.
- L'informatica e gli algoritmi.
- Algoritmi, programmi e linguaggi di programmazione.
- L'informazione e le sue unità di misura: bit, byte e multipli.

1.1 L'informatica e gli algoritmi

L'*informatica* può essere definita come la disciplina che si occupa dell'*informazione* e del suo trattamento in maniera *automatica*.

Questa definizione di informatica evidenzia due elementi fondamentali: da un lato il concetto di *informazione*, dall'altro i *mezzi* per elaborarla. Come mezzi per l'elaborazione dell'informazione, non intendiamo solo i mezzi fisici, come i computer, ma anche, e prima di tutto, i *procedimenti* di elaborazione, che prendono il nome di *algoritmi*.

Il concetto centrale dell'informatica è appunto quello di *algoritmo*, come procedimento per la trasformazione delle informazioni. Lo studio degli algoritmi può essere svolto indipendentemente dagli strumenti fisici utilizzati per la manipolazione dell'informazione, i computer e, in effetti, la sua nascita ha preceduto di molti secoli quella dei computer.

In realtà la nozione di algoritmo non riguarda solo l'informatica e il trattamento dell'informazione, ma riguarda qualunque campo in cui si possano descrivere sequenze di operazioni, finalizzate allo svolgimento di un compito. Ad esempio, una ricetta di cucina è un algoritmo che può essere eseguito da un cuoco, mentre uno spartito musicale è un algoritmo che può essere eseguito da un pianista.

Il primo esempio di algoritmo che presentiamo è in ambito matematico. Consideriamo il problema del calcolo del massimo comun divisore tra due interi positivi. Tale problema può essere risolto con il seguente procedimento, noto come *algoritmo di Euclide*, dal nome del matematico greco che l'ha scoperto:

1. Siano x e y i due numeri.
2. Calcola il resto della divisione di x per y .
3. Se il resto è diverso da zero, ricomincia dal passo 2
utilizzando come x il valore attuale di y , e come y il valore del resto,
altrimenti prosegui con il passo successivo.
4. Il massimo comun divisore è uguale al valore attuale di y .

©1999 Giovanni Pighizzini

Il contenuto di queste pagine è protetto dalle leggi sul copyright e dalle disposizioni dei trattati internazionali. Il titolo ed i copyright relativi alle pagine sono di proprietà dell'autore. Le pagine possono essere riprodotte ed utilizzate liberamente dagli studenti, dagli istituti di ricerca, scolastici ed universitari afferenti ai Ministeri della Pubblica Istruzione e dell'Università e della Ricerca Scientifica e Tecnologica per scopi istituzionali, non a fine di lucro. Ogni altro utilizzo o riproduzione (ivi incluse, ma non limitatamente a, le riproduzioni a mezzo stampa, su supporti magnetici o su reti di calcolatori) in toto o in parte è vietata, se non esplicitamente autorizzata per iscritto, a priori, da parte dell'autore.

L'informazione contenuta in queste pagine è ritenuta essere accurata alla data della pubblicazione. Essa è fornita per scopi meramente didattici e non per essere utilizzata in progetti di impianti, prodotti, ecc.

L'informazione contenuta in queste pagine è soggetta a cambiamenti senza preavviso. L'autore non si assume alcuna responsabilità per il contenuto di queste pagine (ivi incluse, ma non limitatamente a, la correttezza, completezza, applicabilità ed aggiornamento dell'informazione). In ogni caso non può essere dichiarata conformità all'informazione contenuta in queste pagine. In ogni caso questa nota di copyright non deve mai essere rimossa e deve essere riportata anche in utilizzi parziali.

Si osservi che chiunque sappia comprendere ed eseguire le operazioni che costituiscono l'algoritmo di Euclide, può calcolare il massimo comun divisore tra due numeri, senza conoscere il significato dell'algoritmo stesso e, addirittura, senza sapere cosa sia il massimo comun divisore tra due numeri. In altre parole, l'esecutore può eseguire il procedimento senza avere la minima idea di quale sia lo scopo di tale procedimento (possiamo immaginare che l'esecutore si limiti ad eseguire fedelmente gli ordini che gli vengono impartiti, senza chiedersi nulla circa il loro significato). Pertanto l'"intelligenza" necessaria per trovare la soluzione del problema (in questo caso per calcolare il massimo comun divisore) è tutta codificata nell'algoritmo.

Vediamo ora di definire con maggior precisione la nozione di algoritmo: un *algoritmo* è un insieme ordinato di passi *eseguibili e non ambigui*, che definiscono un processo che *termina*.

Osserviamo che nella definizione precedente si richiede che i passi che costituiscono un algoritmo siano *eseguibili* o, in altre parole, che l'algoritmo sia *effettivo*. Si considerino, ad esempio, le seguenti istruzioni:

1. Crea un elenco di tutti i numeri primi.
2. Ordina l'elenco in modo decrescente.
3. Preleva il primo elemento dall'elenco risultante.

Queste istruzioni non descrivono un algoritmo effettivo, in quanto la prima e la seconda istruzione non sono effettivamente eseguibili.

La seconda condizione presente nella definizione di algoritmo è che i passi siano *non ambigui*. Questo significa che l'azione da compiere ad ogni passo deve essere univocamente determinata, senza permettere gradi di libertà.

Infine, si richiede che il processo definito dall'algoritmo *termini*, cioè conduca alla soluzione, in un numero finito di passi.

1.2 Algoritmi, programmi e linguaggi di programmazione

Abbiamo visto che un algoritmo indica un metodo, un procedimento, per la soluzione di un problema. Un *programma* è l'espressione di un algoritmo in un linguaggio che l'esecutore sia in grado di comprendere *senza bisogno di ulteriori spiegazioni*. Dunque possiamo immaginare un algoritmo come un oggetto astratto, concettuale, mentre un programma è un'espressione concreta dell'algoritmo. Lo stesso algoritmo può essere espresso in differenti linguaggi, in base agli esecutori ai quali è destinato. La scrittura del programma è quindi una fase successiva all'individuazione dell'algoritmo risolutivo di un determinato problema.

Per rappresentare concretamente un algoritmo si utilizzano dei blocchi di base, cioè delle istruzioni primitive, che l'esecutore è in grado di comprendere ed eseguire direttamente, e delle regole per combinare tra di loro le istruzioni primitive. L'insieme delle istruzioni primitive, con le regole di combinazione, costituiscono la base dei *linguaggi di programmazione*. Ogni istruzione primitiva ha una *sintassi* e una *semantica*. La *sintassi* è la forma con cui l'istruzione primitiva viene espressa, la *semantica* è il significato dell'istruzione stessa. L'esecutore, riconoscendo un'istruzione in base alla sintassi, effettua l'azione corrispondente alla semantica.

1.3 Informazione

Le entità su cui opera un computer (come i dati e i risultati) prendono il nome di *informazione*. L'informazione può essere misurata, utilizzando un'unità detta *bit* (da *binary digit*). Una trattazione rigorosa di queste tematiche verrà presentata nel corso di *Teoria dell'Informazione*. Intuitivamente, un bit corrisponde alla quantità di informazione che otteniamo quando riceviamo la risposta a una domanda binaria, cioè a una domanda che ammette solo le due risposte "sì" e

“no” (nell’ipotesi che le due risposte abbiano la stessa probabilità). In altri termini, con un bit d’informazione possiamo rappresentare uno tra due possibili valori, come giorno/notte, sole/luna, acceso/spento, falso/vero, 0/1, ecc. Con due bit possiamo rappresentare quattro valori differenti (es. 00, 01, 10, 11). In generale, con n bit, possiamo rappresentare 2^n valori differenti.

Una sequenza di 8 bit viene detta *byte*. Pertanto un byte rappresenta un valore tra $256 = 2^8$ possibili valori o, in altri termini, un byte può assumere 256 valori differenti. I prefissi K (Kilo), M (Mega) e G (Giga) indicano alcuni multipli del byte: 2^{10} (un po’ piú di mille), 2^{20} (un po’ piú di un milione), 2^{30} (un po’ piú di un miliardo) byte, rispettivamente.