Università di Trieste

Laurea in ingegneria elettronica e informatica

Enrico Piccin - Corso di Algoritmi e Strutture dati - Prof. Andrea Sgarro

Anno Accademico 2021/2022 - 2 Marzo 2022

Indice

2 Marzo 2022

1 Introduzione

Algoritmo è una parola molto antica, non connessa all'utilizzo e all'invenzione del calcolatore.
Liber abaci (tradotto "Libro della computazione"), scritto nel 1200 da Leonardo Bonacci, in contatto con la popolazione araba, in quanto mercante ed è venuto a conoscenza della **numerazione araba**, introducendola in Occidente e spiegandolo all'interno del **Liber abaci**.
La numerazione araba è uno straordinario passo in avanti nella scienza, in quanto viene introdotto il concetto di **notazione posizionale**, così come l'importanza del numero 0: i numeri non servono solamente per contare.
A Firenze, sempre negli stessi anni, ci fu una **protesta sindacale** contro l'innovazione tecnologica, contro una nuova scoperta, facendo pressione affinché il governo abolisse il nuovo sistema di numerazione, in quanto avrebbe fatto perdere il posto di lavoro a tutti coloro che prima eseguivano difficili calcoli con la numerazione romana: tuttavia, tale proteste possono rallentare il progresso, ma mai arrestarlo.
Leonardo Bonacci, nei suoi viaggi in Oriente, venne a conoscenza del **Liber abaci** di Al-Gorasmy, proveniente dalla Coresmia, ma che parlava persiano, da cui poi sarebbe stato tratto il nome **Algoritmo**, che letteralmente significa **procedimento di calcolo**.
Tuttavia, bisogna chiarire che algoritmo è un procedimento di calcolo non necessariamente numerico, ma molto più generale, che va ben al di là dei numeri.
Un altro importante elemento che contraddistingue l'algoritmo è òa **meccanicità**, ovvero la sua esecuzione può essere affidata ad una macchina: non deve necessariamente essere meccanizzato, ma deve essere meccanizzabile; in altre parole, l'esecuzione (bada bene, l'esecuzione e non la sua ideazione) dell'algoritmo è completamente **stupida**.

Esempio 1: L'algoritmo della moltiplicazione è molto chiaro e semplice: basta solamente accedere ad una **base di dati** in cui sono memorizzati i prodotti elementari fra numeri molto piccoli, e quindi molto più semplici da trattare.
Una volta eseguite le operazioni di moltiplicazione tramite quanto esposto in precedenza, è necessario eseguire delle operazioni di addizione, che era necessario aver precedentemente memorizzato.
Ecco che quello che si è appena eseguito è un **procedimento di calcolo**.

Esempio 2: L'algoritmo della divisione fa sempre uso di una base di dati, nella quale devono essere memorizzati i risultati del prodotto del divisore con tutti i numeri decimali da 0 a 9 e confrontare ciascun prodotto con il termine da dividere per ottenere il quoziente.
Alternativamente, si sarebbe potuto creare un ciclo da 0 a 9 in cui per ogni indice si sarebbe dovuto verificare se questo fosse il fattore moltiplicativo corretto per ottenere la quantità corretta da sottrarre.

Osservazione: In ciascuno di tali esempi è essenziale la meccanicità del processo esecutivo, che appare evidente.

Quando si rappresentano delle quantità e, a maggior ragione, quando si effettuano dei calcoli, è fondamentale fissare una base di rappresentazione, da cui poi dipendono le cifre che si possono impiegare per la rappresentazione.
La notazione posizionale permette anche di comprendere la rappresentazione di qualsiasi in quantità con qualsiasi base, effettuando anche delle conversioni di base a seconda della maggiore o minore convenienza di rappresentazione.
Per esempio, volendo convertire una quantità rappresentata in base $\mathcal{B} = 7$ in una base $\mathcal{C} = 10$ si deve procedere come segue

$$(5203)_7 = 5 \cdot 7^3 + 2 \cdot 7^2 + 0 \cdot 7^1 + 3 \cdot 7^0 = 1715 + 98 + 0 + 3 = (1816)_{10}$$

Ovviamente le basi di rappresentazione sono almeno binarie, in quanto la **base unaria** non può, per ovvie ragioni, non permette di rappresentare alcuna quantità se non quella unica che viene permessa dalla base scelta, ossia lo 0.
Tuttavia, il processo inverso, atto a passare dalla rappresentazione di una quantità in base 10 ad una in base 3, non risulta essere così immediato.

Per cercare un algoritmmo che permette di effettuare tale conversione, si effettua un primo passaggio controintuitivo, che prevede di rappresentare una quantità in base 10 in una quantità ancora in base 10, tramite un processo di divisioni successive. Si consideri, a tal proposito

$$(3412)_{10}$$

e si divida progessivamente tale numero per 10, come segue

3412	10
341	2
34	1
3	4
0	3

Leggendo, ora, i resti, al contrario si ottiene il numero cercato all'inizio. Se ora si prova a considerare un'altra base, come 3, l'operazione porta ad un risultato analogo

3412	3
1137	1
379	0
126	1
42	0
14	0
4	2
1	1
0	1

Per cui si è ottenuto

$$(3412)_{10} = (11200101)_3$$

Scegliendo la base 2 si ottiene, per esempio

241	2
120	1
60	0
30	0
15	0
7	1
3	1
1	1
0	1

Per cui si è ottenuto

$$(241)_{10} = (11110001)_2$$

Ovviamente la lunghezza di rappresentazione in base 2 prevede un numero di cifre pari a circa il triplo di quelle impiegate per rappresentare la medesima quantità in base 10, proprio perché

$$\log_2(10) \cong 3.3$$

Per passare da base 10 a base 100, le operazioni sono molto semplici

$$(375712)_{10} = [(37)\,(57)\,(12)]_{100}$$

usando come simboli

$$(00)\,,(01)\,,...,(75)\,,...,(99)$$

Si consideri, ora la base 8 e si scriva un numero binario in base ottale:

$$(010101010)_2 = [(010)\,(101)\,(010)]_8 = (252)_8$$

Ancora una volta, le cifre impiegate per la rappresentazione sono state ridotte ad un terzo, sempre perché

$$\log_2(8) = 3$$

E se ora si volesse impiegare la base 16 si otterrebbe:

$$(010101010)_2 = [(1010)\,(1010)]_{16} = (\mathrm{AA})_{16}$$

Osservazione: Si consideri una lunghezza $l = 5$. Allora usando 5 cifre, non tutte nulle, in base 10, i numeri n che si possono rappresentare sono

$$10000 \leq n \leq 99999 \quad \equiv \quad 10^4 \leq n < 10^5 \quad \equiv \quad 10^{l-1} \leq n < 10^l$$

Da ciò si può estrapolare un risultato importante

$$\log_{10}\left(10^{l-1}\right) \leq \log_{10}(n) < \log_{10}\left(10^l\right) \quad \equiv \quad l - 1 \leq \log_{10}(n) < l$$

che è una relazione esatta. Tuttavia, approssimativamente, si può scrivere che

$$l_{10}(n) \cong \log_{10}(n)$$