

LABORATORIO PLS:

‘VERSO IL LICEO MATEMATICO’

- Scopo del laboratorio è discutere con i docenti di matematica e di altre discipline della scuola interessata al laboratorio, e sperimentare con gli studenti del primo anno, un percorso di approfondimento di alcuni argomenti del programma che metta in luce i collegamenti della matematica con le altre materie e sviluppi alcuni punti di vista meno usuali.
- Gli incontri ruoteranno attorno alle due parole chiave: ‘**dimostrazione**’ e ‘**algoritmo**’.

- Nella IDS del Liceo Scientifico G. De Sanctis saranno dedicate 30/32 ore allo svolgimento del Laboratorio divise sui due temi, con alcuni incontri all'Università e alcuni a scuola.
- I docenti della classe impegnati sono la prof.ssa Granese di Geostoria, la prof.ssa Puzio di Matematica e la Prof.ssa Savinelli di Fisica.
- Come per il tema riguardante la dimostrazione, anche per l'algoritmo il lavoro è iniziato a partire da cenni storici e non strettamente matematici, per poi proseguire con esempi di semplici algoritmi.

Dimostrazione e algoritmo



Papiro di Rhind

Algoritmi

Un approccio storico-didattico

The word “algorithm” itself is quite interesting; at first glance it may look as though someone intended to write “logarithm” but jumbled up the first four letters.
Donald E. Knuth, The Art of Computer Programming, v. I, p.1.

Non appena una cosa viene dimostrata impossibile a compiersi, una quantità di gente si mette al lavoro per cercare di realizzarla comunque. Questa sembra una componente essenziale del comportamento umano.
Donald E. Knuth, Algoritmi, “Le Scienze” n. 108, agosto 1977, p. 14.

Algoritmo:

La forma più antica di questo termine si trova nel latino medievale, in cui, con la parola **algorithmus** o **algorismus**, si designava ogni procedimento per eseguire operazioni aritmetiche facendo uso delle cifre arabe che erano state introdotte in occidente con l'opera Liber Abaci, del 1202, di Leonardo Pisano, il grande matematico italiano noto con il soprannome di Fibonacci.



Ritratto di Fibonacci

Nel Medioevo veniva fatta una distinzione tra i cosiddetti **abacisti**, che calcolavano con l'abaco, e gli **algoristi**, che calcolavano usando le nuove cifre arabe.

Nel Rinascimento si riteneva che l'origine della parola fosse dovuta alla combinazione delle parole **algiros** (penoso) e **arithmos** (numero); mentre altri ritenevano che derivasse da “Re Algor di Castiglia”.

Finalmente, gli storici della matematica hanno stabilito che il termine medievale **algorismus** deriva dal nome del matematico persiano Abu ‘Abd Allah Muhammad ibn Musa al-Khwarizmi (c. 825) - letteralmente, “Padre di Abdullah, Mohammed, figlio di Moses, nativo di Khwarizm.”

Khwarizm era una regione dell'Asia centrale, localizzata nel bacino del fiume Amu, a sud del Mar d'Aral che un tempo era noto come Lago Khwarizm.

Al-Khwarizmi scrisse l'opera Kitab Al-jabr wa'l Muqabala (L'arte di numerare ed ordinare le parti in un tutto) da cui deriva il nome **Algebra**.

La traduzione di questo trattato in latino fu fatta molti anni dopo e cominciava con le parole **Dixit Algorithmi**, storpiando il soprannome del suo autore; da qui ebbe origine il termine **algoritmo**.

Fino ad alcuni decenni fa la parola «**algoritmo**» era sconosciuta alla maggior parte delle persone colte, ma lo sviluppo rapidissimo avuto dalla scienza dei calcolatori, nell'ultimo ventennio del ventesimo secolo, ha modificato la situazione, ed attualmente la parola è diventata indispensabile, e le pubblicazioni riguardanti gli algoritmi hanno raggiunto un numero considerevole.

Esistono diversi termini che, almeno in parte, esprimono il concetto in questione, come: procedura, prescrizione, routine, processo, metodo.

Si può definire un **algoritmo un insieme di regole o direttive atte a fornire una risposta specifica a una specifica richiesta.**

Caratteristica distintiva degli algoritmi è la **totale eliminazione delle ambiguità**: le regole devono essere sufficientemente semplici e ben definite da poter essere eseguite da una macchina.

Un'altra caratteristica fondamentale degli algoritmi è che **devono sempre avere termine dopo un numero finito di passi.**

Lo studio sistematico degli algoritmi, la loro definizione matematica, l'analisi delle operazioni da eseguire, la precisazione delle classi di problemi risolubili mediante algoritmi sono stati oggetto di ricerca nel secolo scorso.

A tale riguardo, studi importanti vennero eseguiti da Andrei Andreievic Markov (1856-1922).

All'inizio degli anni trenta venne sviluppata la teoria delle funzioni ricorsive da parte di molti matematici, tra i quali spicca il grande Kurt Gödel.



Kurt Gödel



Alan Turing

Anche la teoria delle macchine 'computazionali', che è strettamente collegata allo studio degli algoritmi, venne sviluppata a partire dagli anni trenta da Alan Turing (1912-1954).

Alcuni esempi di algoritmi

a] Uno degli algoritmi più antichi venne scoperto su alcune tavolette di argilla in Mesopotamia e si riferisce al calcolo della somma delle potenze del numero 2 con esponente variabile da 1 ad n .

Rappresentandolo nel nostro linguaggio, esso si compone dei passi seguenti:

1. Inizio dell'algoritmo.
2. Somma la potenza di 2 da 1 ad n , con n numero intero ≥ 1 .
3. L'ultimo termine della somma è 2^n .
4. Sottrai 2 da 2^n , troverai quindi $2^n - 2$.
5. Somma $(2^n - 2)$ a 2^n e ottieni la risposta; quindi la somma è $S = 2^n + (2^n - 2)$.
6. Questo è il procedimento.

Algebricamente, questo procedimento della matematica babilonese viene convalidato dalla formula che fornisce la somma delle potenze di 2, quando l'esponente k assume tutti i valori interi positivi da 1 ad n :

$$\sum_k 2^k = 2^n + (2^n - 2)$$

Questa formula rappresenta, infatti la somma di n termini della progressione geometrica:

$$2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, \dots$$

di ragione 2, per cui la somma vale:

$$S_n = a_1 \cdot \frac{q^n - 1}{q - 1} = 2 \cdot \frac{2^n - 1}{2 - 1} = 2 \cdot 2^n - 2 = 2^n + (2^n - 2)$$

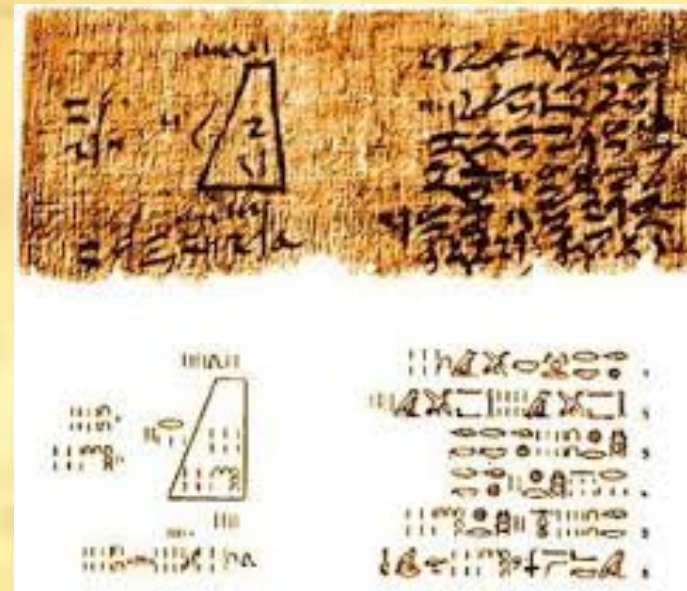
Ciò che appare subito evidente è che **qualsiasi persona, pur non conoscendo il concetto di progressione geometrica, può raggiungere il risultato desiderato**, seguendo le singole istruzioni del procedimento descritto.

L'algoritmo nell'antichità

L'algoritmo utilizzato in matematica venne adottato già dagli Egizi. Ne siamo venuti a conoscenza grazie a due papiri, il Papiro di Rhind e il Papiro di Mosca.



Papiro di Rhind



Papiro di Mosca

Papiro di Rhind

Contiene al suo interno alcuni problemi geometrici di volume, una rappresentazione del teorema di Pitagora e vari tipi di frazione. È la fonte più ampia che abbiamo sulla matematica egizia. Risale al Regno Medio fra il 2000 e il 1800 a.C.

Papiro di Mosca

Presenta la formula del volume di una piramide di base quadrata e un metodo per ricavare l'area di un emisfero. Fu scritto da un ignoto scriba circa nel 1890 a.C. Contiene anche venticinque esempi di vita quotidiana.

b] Nel famoso Papiro Rhind, che si ritiene scritto dallo scriba egizio Ahmes, circa nel 1650 a.C., e conservato nel British Museum di Londra, spicca un algoritmo di moltiplicazione tra interi che coincide, sorprendentemente, con il nostro usuale algoritmo di moltiplicazione, se gli operandi fossero rappresentati in base 2; solo che gli egiziani non conoscevano tale rappresentazione!



Una parte del Papiro Rhind

Tale algoritmo può essere descritto mediante i passi seguenti.

- 1] Sono dati i numeri interi positivi a e b . Sia p il loro prodotto, inizialmente incognito.
- 2] Poni $p = 0$ e controlla se a è dispari o pari.
- 3] Finché $a \neq 0$ ripeti la seguente procedura:
 - 3.1 se a è dispari, allora fai la somma di b e di p ;
 - 3.2 Dimezza a trascurando il resto;
 - 3.3 Raddoppia b .
- 4] Scrivi il risultato finale.

Per esempio se $a = 45$ e $b = 14$, il prodotto $p = 630$ si costruisce come segue:

$p :$	0	14	14	70	182	182	630
$a :$	45	22	11	5	2	1	0
$b :$	14	28	56	112	224	448	896

In questo esempio, se rappresentiamo 45 in base 2 otteniamo:

1 0 1 1 0 1

Scriviamo questo numero in forma polinomiale:

$$2^5 + 2^3 + 2^2 + 2^0$$

Se eseguiamo ora la moltiplicazione per 14, si ha:

$$(2^5 + 2^3 + 2^2 + 2^0) \cdot 14 = 448 + 112 + 56 + 14 = 630$$

che è la somma corrispondente al prodotto $45 \cdot 14$.

Un procedimento simile potremmo utilizzarlo per fare eseguire la moltiplicazione tra due numeri anche ad un allievo che conosce i simboli 1, 2, 3, ..., che sa sommare e sottrarre, che possiede fogli di carta, ma non sa che cosa significhi «moltiplicare». Potremmo, per esempio, descrivere l'operazione nel modo seguente, pur sapendo che il metodo descritto sarebbe lontano dalla nostra idea di «moltiplicazione»:

Inizio

- scrivi il numero a sul foglio A

- scrivi il numero b sul foglio B

- scrivi il numero 0 sul foglio C

- finché sul foglio B c'è un numero maggiore di 0 ripeti le seguenti istruzioni:

 - inizio

 - fai la somma del numero scritto sul foglio C col numero scritto sul foglio A

 - scrivi il risultato sul foglio C dopo aver cancellato il numero precedente

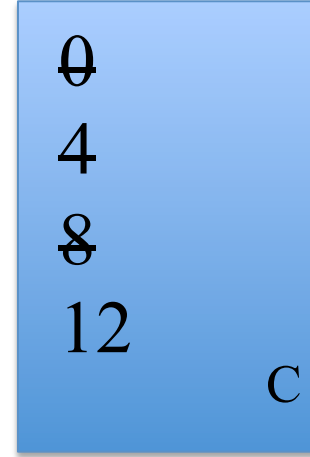
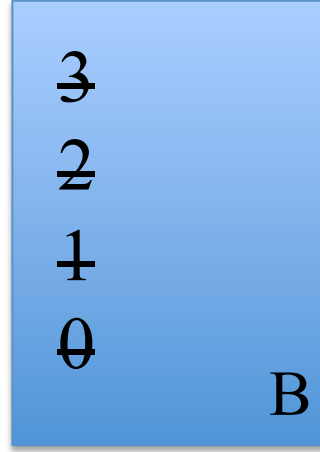
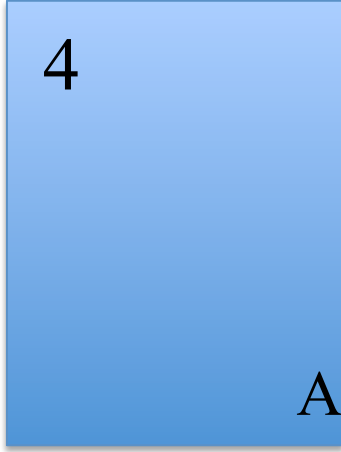
 - sottrai 1 al numero scritto sul foglio B

 - scrivi il risultato sul foglio B dopo aver cancellato il numero precedente

 - fine

- leggi il numero scritto sul foglio C

fine.



c] Un altro calcolo guidato si trova nel Papiro di Mosca, risalente al 1700 a.C., in cui sono risolti 25 problemi della stessa natura di quelli del Papiro Rhind.

Uno dei più interessanti è il seguente, in cui si calcola **il volume di un tronco di piramide a base quadrata** secondo questo metodo:

Se ti viene detto: c'è una piramide tronca che ha 6 per altezza verticale, 4 per la base e 2 per la cima.

Fai il quadrato di 4, risultato 16.

Raddoppia 4, risultato 8.

Fai il quadrato di 2, risultato 4.

Addiziona il 16, l'8 e il 4, risultato 28.

Poi prendi un terzo di 6, risultato 2.

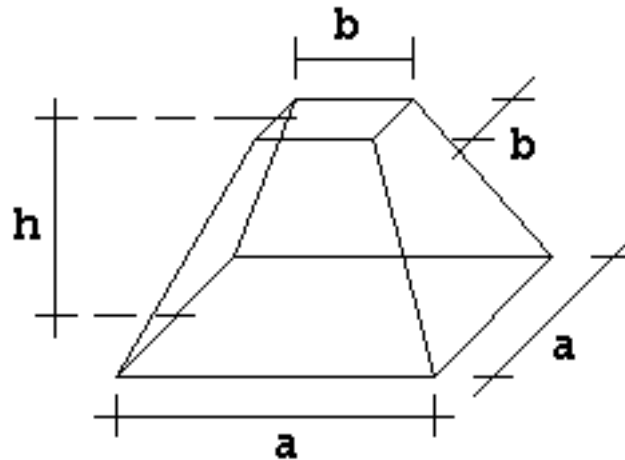
Allora fai due volte 28, risultato 56.

Vedrai che il risultato è giusto.

Il problema del tronco di piramide nel Papiro di Mosca

L'algoritmo si compone, quindi, dei seguenti passi:

1. Inizio dell'algoritmo (altezza $h=6$, base $a=4$, cima $b=2$).
2. Fai il quadrato di 4.
3. Scrivi il risultato: 16.
4. Fai il doppio di 4.
5. Scrivi il risultato: 8.
6. Fai il quadrato di 2.
7. Scrivi il risultato: 4.
8. Somma $16 + 8 + 4$.
9. Scrivi il risultato: 28.
10. Calcola $\frac{1}{3}$ di 6.
11. Scrivi il risultato: 2.
12. Moltiplica 2 per 28.
13. Scrivi il risultato: 56.
14. Fine dell'algoritmo.



Ciò che meravaglia maggiormente in questi esempi è che essi conducono il lettore al risultato giusto, ma **senza motivare la successione delle operazioni**.

Essi appaiono incredibilmente come algoritmi il cui esecutore è pensato più come un automa che come un essere umano.

Nel caso del tronco della piramide **non viene fornito un metodo generale valido** per tutti i tipi di questi solidi.

Non si può sperare, in questo tipo di geometria pratica, di trovare una formula generale come quella che usiamo noi per trovare il volume di qualsiasi tronco di piramide, conoscendo il lato **a** della base maggiore, il lato **b** della base minore e l'altezza **h**:

$$V = \frac{1}{3} h(a^2 + ab + b^2)$$

L'unica indicazione è quell'affermazione finale: “vedrai che il risultato è giusto”, che sembra voler dire: “non preoccuparti, è come ti dico io”, senza che il lettore possa chiedere: “ma perché si deve fare così?”

Nell'uso odierno, per algoritmo si intende, sinteticamente, qualunque schema o procedimento di calcolo; più precisamente:

Un procedimento di calcolo esplicito e descrivibile con un numero finito di regole che conduce al risultato dopo un numero finito di operazioni, cioè di applicazione delle regole.

Anche se il concetto di algoritmo ha una connotazione precisa solo nella scienza, e in particolare nella matematica, esso non è un patrimonio solo dell'ambito scientifico, ma sta alla base di molte azioni che di solito vengono dette abitudinarie, ma che possono essere decodificate mediante successioni di atti che il nostro cervello ha ormai immagazzinato in maniera definitiva e che, ripetute nel loro complesso, appaiono naturali.

Tali sono, per esempio, le azioni che riguardano:

- aprire una porta;
- prendere l'ascensore;
- fare una telefonata;
- mettere in moto l'automobile.

d] Consideriamo, ad esempio, l'azione di “**fare una telefonata**”. Alla base dell'azione c'è la necessità di **risolvere un problema**: comunicare qualcosa ad un'altra persona che è irraggiungibile fisicamente, e che vive nella nostra stessa città o in un'altra città.

Ci avviciniamo all'apparecchio telefonico, alziamo il ricevitore, componiamo il numero e attendiamo che la persona chiamata ci risponda. Se risponderà, allora avremo risolto il nostro problema; in caso contrario, potremo decidere di riprovare dopo qualche tempo.

Se analizziamo la successione di azioni che si devono compiere, possiamo elencarle nel modo seguente:

- 1] mi avvicino all'apparecchio telefonico;
- 2] sollevo il ricevitore;
- 3] attendo il segnale acustico;
- 4] compongo il numero;
- 5] se il numero chiamato è libero, attendo la risposta, altrimenti riaggancio il ricevitore.

Chiediamoci: a quali criteri soddisfa questa sequenza di azioni?

essa è finita, cioè, termina in un tempo finito;

non crea contraddizioni, ovvero, non mi porta a fare un'azione che sia in contrasto con un'azione fatta precedentemente nella sequenza;

fa sempre la stessa cosa, cioè, ogni volta che io ripeterò la successione delle azioni , come risultato finale potrò comunicare con la persona chiamata oppure no.

Possiamo concludere che **questo algoritmo è buono**, ovvero risolve efficacemente il nostro problema.

e] Ecco un altro problema:

Devo spiegare ad un amico come arrivare a casa mia, partendo da un posto noto ad entrambi.

Dopo avergli disegnato una piantina sommaria del luogo da raggiungere, potrei fornirgli queste indicazioni:

- 1] partendo dal punto noto, prendi la prima strada a destra;
- 2] prosegui lungo questa via fino alla seconda trasversale;
- 3] gira a sinistra;
- 4] prosegui fino a che vedi sulla tua destra una bella insegna di negozio;
- 5] gira a sinistra;
- 6] prosegui lungo questa via fino al n. 33.

A prima vista, sembra che la successione delle istruzioni vada bene e che l'algoritmo debba funzionare.

In realtà c'è qualcosa che non lo rende del tutto efficace: precisamente il passo 4, quando viene detto all'amico di proseguire finché non vede una bella insegna!

In tal caso ho ipotizzato che quella che per me è una bella insegna debba esserlo pure per il mio amico!

Ciò che può anche non essere, perché il concetto di “bello” è soggettivo.

Quindi **questo algoritmo non è buono.**

f] Consideriamo un altro problema, questa volta di natura matematica: Devo **determinare l'area di un rettangolo**, conoscendo le sue dimensioni **a** e **b**:

In questo caso i passi da compiere sono:

- 1] scrivo la formula per determinare l'area del rettangolo: **$A = a \cdot b$** ;
- 2] sostituisco ad **a** e a **b** i dati numerici;
- 3] moltiplico i valori numerici di **a** e di **b** (moltiplicando indifferentemente a per b oppure b per a);
- 4] scrivo il valore ottenuto.

Anche in questo caso possiamo affermare che **l'algoritmo usato è efficace**, in quanto si conclude in un tempo finito, non conduce ad alcuna contraddizione, si può applicare a problemi simili, permettendomi di risolverli.

Ma non è il solo algoritmo che mi permetta di risolvere questo problema.

Considero il rettangolo dato come ottenuto dalla somma di un quadrato di lato **b** e di un rettangolo di lati **b** e **a - b**.

La sua area sarà:

$$A = b^2 + b(a - b)$$

Nello stesso modo, l'area del rettangolo ottenuto sarà data dalla somma di un quadrato il cui lato è **a - b** e di un rettangolo di lati **a - b** e **2b - a**:

$$A = b^2 + (a - b)^2 + (a - b)(2b - a)$$

Continuando nello stesso modo, otterremmo per l'area del rettangolo, una successione di quadrati e di rettangoli che non avrà mai fine, o che mi farà raggiungere il risultato dopo una lunga successione di operazioni, per cui **l'algoritmo usato non potrà ritenersi buono.**

g] Consideriamo ora il seguente problema:

Ottenere il numero 13 iniziando dal numero zero.

Supponiamo di volere risolvere il problema mediante il seguente algoritmo:

- 1] attribuisco ad A il valore 0;
- 2] aggiungo 2 ad A;
- 3] scrivo il valore di A;
- 4] ripeto i passi 2 e 3 finché A diventa uguale a 13.

Eseguendo le istruzioni, questo algoritmo fornisce la successione di valori:

0, 2, 4, 6, 8, 10, 12, 14, 16, . . .

Esso dunque non risolve il problema né ha termine, per cui **non è un buon algoritmo.**

h] Consideriamo, ora, un algoritmo per ordinare un insieme di elementi, siano essi numeri o parole.

Per capire come costruire l'algoritmo, supponiamo che i numeri siano tre: 5, 3, 2.

Il primo passo consisterà nel confrontare il primo numero con il secondo ed effettuare uno scambio nel caso in cui il primo sia maggiore del secondo. Otteniamo:

3, 5, 2

Ora confrontiamo il primo numero con il terzo ed effettuiamo lo scambio dopo il confronto, qualora sia necessario. Otteniamo:

2, 5, 3

Ora confrontiamo il secondo numero con il terzo, ed effettuiamo lo scambio, se necessario. Otteniamo:

2, 3, 5.

Abbiamo finito, perché i tre numeri ora sono ordinati.

I passi dell'algoritmo potrebbero essere, allora, i seguenti:

1. Siano a , b , c i tre numeri dati;
2. Se $a > b$, allora scambia a con b , altrimenti non fare nulla;
3. Se $a > c$, allora scambia a con c , altrimenti non fare nulla;
4. Se $b > c$, allora scambia b con c , altrimenti non fare nulla;
5. Scrivi l'ultima sequenza dei numeri.

i] Consideriamo ora un esempio di algoritmo di natura diversa dai precedenti, relativo al “gioco dell'undici”.

Su di un tavolo vengono posti undici oggetti, uguali o diversi.

Due giocatori, A e B, devono raccogliere a turno da uno a tre degli oggetti.

Perde chi raccoglie l'ultimo oggetto.

Ebbene, se il giocatore A deve fare la prima mossa, allora egli potrà essere sicuro di vincere se seguirà le seguenti istruzioni di gioco:

1. A raccoglie 2 oggetti.
2. Il gioco passa al giocatore B.
3. B raccoglie k oggetti, con $1 \leq k \leq 3$.
4. Finché sul tavolo ci sono oggetti, A raccoglie $4 - k$ oggetti e si torna al passo 2.
5. A vince.

Questo algoritmo è stato portato come esempio di quella che dev'essere l'**efficacia di un algoritmo**.

In questo caso l'algoritmo **risolve una classe di problemi e non un singolo problema**, perché i problemi sono tanti quante sono le possibili mosse del giocatore B.

Infatti, B può prendere 1, 2 o 3 oggetti, ogni volta che viene il suo turno.

Quindi, potremmo dire che **risolvere un problema** significa ricercare ed esprimere una **successione finita di istruzioni**, interpretabili da un esecutore, che conducano da determinate informazioni iniziali ad altre informazioni finali.

Infatti, che cosa facciamo quando tentiamo di risolvere un problema?

Procediamo sempre allo stesso modo: ci serviamo di un **modello**, ovvero di una rappresentazione semplificata dello stesso problema, sfrondandolo di tutto ciò che è superfluo per raggiungere l'obiettivo (cioè la soluzione) che ci si propone.

Dunque, alla base della costruzione di un algoritmo vi è la necessità di **rendere operativo il modello** adottato per risolvere il problema.