Dario Palladino[1]

# Introduzione al concetto di algoritmo

*Con il recente enorme sviluppo delle discipline informatiche e l'uso sempre più diffuso dei calcolatori, il concetto di algoritmo ha assunto un ruolo centrale in ambito scientifico. Fin dall'antichità gli algoritmi hanno accompagnato lo sviluppo della matematica, ma solo nel secolo scorso sono stati fatti oggetto di indagine diretta, dando origine ad una nuova teoria matematica, detta Teoria della computabilità. I primi elementi di questa teoria figurano nei programmi di matematica della scuola secondaria. In questo articolo espongo alcuni aspetti relativi agli algoritmi che possono essere introdotti già nel primo biennio della scuola secondaria di secondo grado.*[2]

## PREMESSA

Intuitivamente, un algoritmo è un procedimento di calcolo che si basa sull'applicazione di un numero finito di regole (o istruzioni) che determinano in modo meccanico tutti i singoli passi del procedimento stesso. Il termine "algoritmo" deriva dal nome del matematico arabo Muhammad ibn Mūsa, detto al-Khuwāritzmī (secolo IX d.C.), che fu latinizzato in Algoritmus. L'individuazione di algoritmi ha accompagnato la storia di tutti i settori della matematica, in quanto l'esistenza delle soluzioni di un qualunque problema (se vale o meno una proposizione, se esistono o meno enti che soddisfano determinate condizioni, ecc) è sempre stata accompagnata dalla ricerca di procedimenti per determinare effettivamente tale soluzione. Più recentemente, il termine "algoritmo" è stato esteso a indicare ogni procedimento che consente di risolvere in modo meccanico un qualsiasi problema, relativo anche a enti non matematici, mediante l'applicazione di un sistema esplicito di regole effettive. La ricerca di algoritmi ha assunto particolare rilevanza ai giorni nostri in quanto i calcolatori sono essenzialmente esecutori di algoritmi[3], ed è quindi opportuno trattare l'argomento nell'insegnamento della matema-

[1] Docente di logica alla Facoltà di Lettere e Filosofia dell'Università di Genova, ora in pensione.

[2] Questo articolo costituisce una versione rielaborata della conferenza "Il concetto di algoritmo", tenuta dall'autore al Liceo Scientifico Statale "G. Ferraris" di Varese il 4 novembre 2011, rivolta prevalentemente a studenti del primo biennio.

[3] Va tenuto presente che tutto ciò che si realizza in un calcolatore (filmati, testi, diagrammi, musica, collegamenti, e così via), nonostante la grande e variegata molteplicità dei compiti che si possono eseguire, è *sempre* riconducibile all'esecuzione di un algoritmo. I "programmi" non sono altro che algoritmi scritti in un linguaggio che consente di implementarli al calcolatore.

tica, come peraltro previsto dalle recenti indicazioni ministeriali. Come quasi sempre accade nelle discipline scientifiche, è opportuno procedere per gradi e, in questo articolo, mi propongo di illustrare come l'argomento possa essere introdotto già nel primo biennio della scuola secondaria di secondo grado, in modo che possa essere approfondito negli anni successivi. Si tratta di una tematica che nasconde alcune insidie poiché, come spesso accade quando si definisce rigorosamente un concetto, non sempre quanto si ottiene è in sintonia con le conoscenze di senso comune.

## DECIDIBILITÀ E CALCOLABILITÀ

Le nostre argomentazioni si svolgeranno tutte all'interno dell'aritmetica, ossia nell'ambito della teoria dei numeri naturali, il cui insieme è denotato con **N**, per cui, salvo esplicita indicazione contraria, con "numero" intendiamo "numero naturale"[4].

Fin dalla prima infanzia abbiamo iniziato a familiarizzare con la rappresentazione decimale dei numeri:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,...,

e a definire e studiare le loro *proprietà* (ad esempio, "essere pari", "essere dispari", "essere primo", "essere divisibile per 3"), le *relazioni* fra di essi (ad esempio, "essere minore", "essere maggiore" "essere divisore", "essere compreso fra"), le *funzioni* e *le operazioni* definite in **N** (ad esempio, "elevamento al quadrato", "addizione", "moltiplicazione", "divisione (con resto)", "sottrazione", "radice quadrata").

Per entrare subito nel vivo del tema di questo articolo introduciamo le due seguenti definizioni:

(a) Una proprietà o una relazione aritmetica è detta **decidibile** se e solo se esiste un procedimento meccanico che consente di stabilire se essa sussiste o non sussiste.

(b) Una funzione o una operazione aritmetica è detta **calcolabile** (o **computabile**) se e solo se esiste un procedimento meccanico che consente di calcolarne i valori in corrispondenza dei loro argomenti per i quali tali valori esistono[5].

Le proprietà e le relazioni prima citate sono decidibili, e le funzioni e operazioni prima citate sono calcolabili. Infatti, quando vengono definite, sono subito accompagnate dalla descrizione del procedimento con cui stabilire se sussistono (nel caso delle proprietà e delle relazioni) o con cui calcolare i valori (nel caso delle funzioni e delle operazioni).

Ad esempio, in **N** vale la seguente proposizione:

*comunque dati due numeri a e b, esistono e sono unici due numeri q e r, con r < b, tali che*:

$a = b \times q + r$ (con $r < b$)

[4] Lo studio degli algoritmi viene condotto con riferimento ai numeri naturali poiché gli algoritmi eseguiti su enti non numerici si possono in generale ricondurre ad algoritmi numerici mediante opportune "codifiche", ottenute associando numeri naturali a enti di natura qualsiasi. Ad esempio, tutto ciò che viene realizzato da un calcolatore è frutto di complesse elaborazioni di natura algoritmica eseguite su lunghissime sequenze finite di 0 e 1 (bit), che possono essere concepite come rappresentazioni dei numeri naturali in forma binaria (in base due), anziché nella usuale forma decimale (in base dieci).

[5] Come vedremo più in particolare nel seguito, si considerano anche operazioni non sempre eseguibili. Ad esempio, in **N**, la sottrazione $a - b$ è eseguibile se e solo se $a \geq b$, la radice quadrata di un numero esiste se e solo se il numero è un quadrato perfetto.

*q* è detto il *quoziente* e *r* il *resto* della *divisione* di *a* per *b*.

Molto probabilmente lo studente non ha mai letto, o non ricorda di averlo fatto, l'enunciato precedente. Nella scuola primaria, l'insegnante avrà detto: "Vediamo come si esegue la divisione fra 111 e 15 e si calcolano il quoziente e il resto della divisione" e illustrato il procedimento con cui si calcolano i due numeri 7 (quoziente) e 6 (resto) tali che:

$111 = 15 \times 7 + 6$

Così, dopo aver definito la relazione "essere divisore" fra due numeri: "si dice che *a* è *divisore* di *b* (*b* è *multiplo* di *a*) se e solo se esiste un numero *k* tale che $b = a \times k$", si afferma subito che, per stabilire se *a* è divisore di *b*, si divide *b* per *a*: se il resto *r* è 0, allora *a* è divisore di *b* (*b* è multiplo di *a*, *b* è divisibile esattamente per *a*), altrimenti, se $r \neq 0$, *a* non è divisore di *b*.

Probabilmente uno studente ricorda e sa eseguire i procedimenti con cui ha imparato a eseguire le operazioni, anche se può avere qualche difficoltà a ricordare con precisione le definizioni, ossia sa eseguire la divisione di 111 per 15, senza prima rispondere alla domanda "Che cosa vuol dire dividere 111 per 15?".

È in ogni caso concettualmente importante distinguere i concetti matematici (proprietà, relazioni, funzioni, operazioni) dai procedimenti con cui si stabilisce la loro decidibilità o calcolabilità.

Ad esempio, affermare "3 è divisore di 135" significa che esiste un numero che moltiplicato per 3 dà come prodotto 135; quando si esegue la divisione di 135 per 3 e si ottiene come resto 0, si stabilisce che è vera la proposizione "3 è divisore di 135" ("135 è multiplo di 3", "135 è divisibile esattamente per 3"). Lo stesso risultato, però, si può ottenere con un altro procedimento più rapido, applicando il criterio di divisibilità per 3 e osservando che la somma delle cifre del numero, ossia 1 + 3 + 5 = 9, è multipla di 3, oppure con un procedimento più lungo che consiste nello scrivere in sequenza i multipli di 3 fino ad ottenere 135:

3, 6, 9, 12, 15,..., 129, 132, 135

Pertanto, la decidibilità della proprietà "avere 3 come divisore" ("essere divisibile per 3") e, più in generale, quella della relazione "essere divisore" può essere stabilita con procedimenti diversi.

Il massimo comun divisore di due numeri *a* e *b*, MCD(*a*, *b*), è definito come il numero più grande che è divisore sia di *a*, sia di *b*. Il procedimento con cui si calcola abitualmente MCD(*a*, *b*) consiste nello scomporre *a* e *b* in fattori primi e moltiplicare i fattori comuni alle due scomposizioni presi con il minimo degli esponenti con cui figurano in esse. L'esistenza di tale procedimento consente di affermare che l'operazione MCD di massimo comun divisore è calcolabile. Vedremo più avanti un altro modo per calcolare l'MCD.

## IL CONCETTO DI ALGORITMO

Un algoritmo non è altro che un procedimento meccanico per stabilire la decidibilità di una proprietà o una relazione, oppure la calcolabilità di una funzione o di una operazione. Conosciamo quindi molti algoritmi dato che, come si è detto, tutte le proprietà e relazioni aritmetiche e tutte le funzioni e operazioni aritmetiche che abbiamo incontrato nei nostri studi sono rispettivamente decidibili e calcolabili. Nel caso della decidibilità l'algoritmo conduce a rispondere alla domanda se sussiste o meno una data proprietà o una data relazione, nel caso della calcolabilità l'algoritmo conduce al valore cercato (se esiste) di una data funzione o di una

operazione[6]. L'attributo "meccanico" significa che, in entrambi i casi, l'esecuzione dell'algoritmo consiste nell'applicare un ben preciso insieme di istruzioni in modo che non siano richieste alcuna abilità o intelligenza, ma solo particolare attenzione per non commettere errori nel corso del procedimento[7].

Dall'esame degli algoritmi che conosciamo, si perviene alla seguente definizione:

> Un ***algoritmo*** è un procedimento di calcolo costituito da un insieme di istruzioni. Tali istruzioni fanno uso di un insieme finito di operazioni elementari, le quali si possono assumere come note e primitive (ad esempio, addizione, moltiplicazione, divisione). Le istruzioni devono essere tali che, per poterle applicare, basti saper eseguire le operazioni elementari

e a enunciare le seguenti caratteristiche:

(1) l'insieme delle istruzioni deve essere finito;
(2) la soluzione, se esiste, deve poter essere ottenuta mediante un numero finito di applicazioni delle istruzioni;
(3) all'inizio del calcolo, e ogni qual volta sia stata eseguita un'istruzione, si deve sempre sapere in maniera precisa quale istruzione va eseguita al passo successivo, e quindi non devono esserci due istruzioni diverse che possono essere applicate nello stesso momento. Un procedimento che goda di questa proprietà è detto *deterministico;*
(4) deve essere sempre chiaro se si è giunti o meno al termine del procedimento, e se sono stati ottenuti i risultati desiderati.

## L'ALGORITMO EUCLIDEO DELLE DIVISIONI SUCCESSIVE

Uno degli algoritmi più famosi risale all'antichità ed è un procedimento meccanico per calcolare il massimo comun divisore di due numeri. Esso è illustrato nell'opera matematica più importante della storia, gli *Elementi* di Euclide del III sec. a.C., viene detto "algoritmo euclideo delle divisioni successive", ed è esplicitamente menzionato nei programmi di matematica. Lo illustriamo nei particolari in modo che si possa controllare che esso possiede le caratteristiche enunciate nel paragrafo precedente.

Se uno studente deve calcolare MCD(2 079, 987) o MCD(16 337, 14 911), probabilmente ricorrerà alla scomposizione dei numeri dati in fattori primi e calcolerà l'MCD come prodotto dei fattori comuni presi con il minimo esponente:

---

6 In realtà, anche le operazioni sono funzioni: le operazioni di addizione, moltiplicazione, elevamento a potenza sono funzioni a due argomenti, ossia associano un numero (il risultato dell'operazione, ossia la somma, il prodotto, la potenza) a ogni "coppia" di numeri (addendi, fattori, base ed esponente). Le operazioni di elevamento al quadrato e di estrazione di radice quadrata sono funzioni a un argomento, ossia associano un numero a un singolo numero. Nel caso della radice quadrata, essa esiste solo se il numero radicando è un quadrato perfetto, e questo spiega come mai abbiamo scritto tra parentesi "se esiste". Pertanto, si può parlare solo di "funzioni", anziché di "funzioni e operazioni".

7 Proprio perché non è richiesta alcuna capacità intellettiva, l'esecuzione di algoritmi può essere affidata a macchine opportunamente programmate, che sono più affidabili dell'uomo poiché non si "distraggono". Ciò che rende sempre più insostituibile il ricorso ai calcolatori è la grandissima velocità con cui eseguono gli algoritmi, in continua rapida crescita con gli sviluppi delle tecnologie.

| | |
|---|---|
| MCD(2 079, 987) | MCD(16 337, 14 911) |
| $2\,079 = 3^3 \times 7 \times 11$ | $16\,337 = 17 \times 31^2$ |
| $987 = 3 \times 7 \times 47$ | $14\,911 = 13 \times 31 \times 37$ |
| $\text{MCD}(2\,079, 987) = 3 \times 7 = 21$ | $\text{MCD}(16\,337, 14\,911) = 31$ |

La scomposizione in fattori primi di un numero è ottenibile mediante un algoritmo, la cui esecuzione richiede parecchio tempo se tra i fattori primi vi è più di un numero per il quale non disponiamo di un criterio di divisibilità, come accade nel caso dei numeri degli esempi precedenti se operiamo solo con carta e penna. Esso richiede l'esecuzione di divisioni per i successivi numeri primi (2, 3, 5, 7, 11, 13,...), le quali consentono di individuare i divisori del numero. Se il numero da scomporre è molto elevato, dell'ordine di centinaia di cifre, e i suoi fattori primi sono grandi, il procedimento può superare le capacità degli attuali calcolatori[8].

L'algoritmo euclideo consente di determinare l'MCD di due numeri senza ricorrere alla scomposizione in fattori primi, ma eseguendo delle divisioni successive: si divide il più grande dei due numeri[9] per l'altro. Se il resto è 0, l'MCD cercato è il più piccolo dei due numeri[10]. Se il resto non è 0, si divide il numero più piccolo per tale resto: se il nuovo resto è 0, il resto precedente è l'MCD cercato, se è diverso da 0 si divide il primo resto per il secondo resto. Si procede con le divisioni fra i resti successivi finché non si ottiene come resto 0. In tal caso il procedimento termina e l'MCD cercato è l'ultimo resto ottenuto diverso da 0. Per chiarire la descrizione del procedimento, applichiamolo nei due esempi precedenti:

| | |
|---|---|
| $2\,079 = 987 \times 2 + 105$ | $16\,337 = 14\,911 \times 1 + 1\,426$ |
| $987 = 105 \times 9 + 42$ | $14\,911 = 1\,426 \times 10 + 651$ |
| $105 = 42 \times 2 + 21$ | $1\,426 = 651 \times 2 + 124$ |
| $42 = 21 \times 2 + 0$ | $651 = 124 \times 5 + 31$ |
| $\text{MCD}(2\,079, 987) = 21$ | $124 = 31 \times 4 + 0$ |
| | $\text{MCD}(16\,337, 14\,911) = 31$ |

L'algoritmo euclideo si basa sostanzialmente sull'unica istruzione di eseguire una divisione, che va ripetuta fino a che non si ottiene come resto 0[11]. Dato che l'MCD di due numeri esiste sempre, lo si ottiene con un numero finito di divisioni. Infatti, i resti via via ottenuti sono decrescenti e quindi, prima o poi, si perviene a un resto nullo e il procedimento, che è evidentemente deterministico, giunge al termine. Che, procedendo in tal modo, si ottenga proprio l'MCD dei numeri dati si basa sul seguente teorema:

*se* $a = bq + r$, *allora* $\text{MCD}(a, b) = \text{MCD}(b, r)$

---

8 Se un numero di circa quattrocento cifre è il prodotto di due numeri primi di circa duecento cifre, è attualmente impossibile individuare i due fattori in un tempo ragionevole, anche utilizzando i più potenti calcolatori oggi disponibili. Su questo fatto si basano, ad esempio, la sicurezza delle transazioni in denaro e la segretezza delle informazioni che navigano in Internet. Lo studio degli algoritmi, quindi, non è solo un fatto interno alla matematica, ma ha anche un enorme interesse applicativo nella vita quotidiana.

9 Se i due numeri sono uguali il loro MCD è ...

10 Infatti, se uno dei due numeri è multiplo dell'altro, il loro MCD è ...

11 Si rilegga la precedente definizione di algoritmo: la divisione è l'operazione elementare assunta come primitiva. Se si descrivesse l'algoritmo con cui si esegue la divisione, nelle sue istruzioni comparirebbero altre operazioni elementari assunte come primitive.

L'idea alla base dell'algoritmo euclideo è semplice: anziché calcolare l'MCD di $a$ e $b$, si divide $a$ per $b$, si trova il resto $r$ e si calcola l'MCD di $b$ e $r$, che sono numeri più piccoli: anziché calcolare MCD(2 079, 987) o MCD(16 337, 14 911), si calcola MCD(987, 105) o MCD(14 911, 1 426).

Iterando il procedimento, i resti via via calcolati sono decrescenti e, dopo un numero finito di divisioni, si trova necessariamente un resto uguale a 0 (una sequenza decrescente di numeri naturali non può procedere all'infinito). A questo punto il procedimento termina in quanto ci si è ricondotti al calcolo dell'MCD di due numeri di cui uno è multiplo dell'altro (negli esempi 42 e 21 oppure 124 e 31), per cui l'MCD cercato è il più piccolo dei due.

*Osservazione*. Il teorema precedentemente enunciato garantisce che si pervenga al risultato desiderato, ossia alla individuazione dell'MCD dei due numeri. È importante tener presente che un algoritmo va sempre accompagnato dalla verifica della sua correttezza, ossia dalla dimostrazione che, eseguendolo, si ottiene quanto richiesto. In molti casi la correttezza dell'algoritmo è del tutto evidente, in altri no. Nel caso dell'algoritmo basato sulla scomposizioni in fattori primi, la correttezza si può dare per scontata (la scomposizione in fattori primi ci fa trovare tutti i divisori di un numero e ci consente di determinare il divisore più grande comune ai due numeri), nel caso dell'algoritmo euclideo delle divisioni successive la correttezza è meno ovvia, in quanto si basa sul teorema prima enunciato, la cui verità non è immediatamente riconoscibile[12]. Quando acquistiamo o scarichiamo un programma per il nostro calcolatore ci aspettiamo che sia stata eseguita la verifica della correttezza del programma stesso.

## ALGORITMI PER LA RADICE QUADRATA

Alcune interessanti considerazioni sugli algoritmi emergono se si considera l'operazione di estrazione della radice quadrata, la quale, come si è osservato, non è sempre eseguibile: affinché esista la radice quadrata di un numero, occorre che esso sia un quadrato perfetto.

Ad esempio, in **N**:

$\sqrt{16} = 4$ $\quad$ $\sqrt{121} = 11$

$\sqrt{2}$ $\quad$ $\sqrt{39}$ $\quad$ $\sqrt{101}$ $\quad$ non esistono[13]

[12] Non è questa la sede per affrontare il problema di come si stabilisce la verità delle proposizioni matematiche. Ci basta ricordare che la verità di quelle non evidenti si può ottenere mediante una dimostrazione. La correttezza dell'algoritmo euclideo delle divisioni successive si basa sulla dimostrazione del teorema precedentemente enunciato. Senza entrare in particolari, a titolo di curiosità, si può procedere come segue dimostrando che, se $a = bq + r$, i divisori comuni di $a$ e $b$ coincidono con i divisori comuni di $b$ e $r$. Infatti, se $k$ è divisore comune di $a$ e $b$, allora $a = mk$ e $b = nk$. Da $a = bq + r$ segue che $r = a – bq = mk – nkq = k(m – nq)$, e quindi $k$ è divisore di $r$, per cui è $k$ è divisore comune di $b$ e $r$. Se $h$ è divisore di $b$ e $r$, allora $b = mh$ e $r = nh$ e si ha $a = mhq + nh = h(mq + n)$, da cui segue che $h$ è divisore di $a$ e quindi è divisore comune di $a$ e di $b$. Dato che $a$ e $b$ hanno gli stessi divisori comuni di $b$ e $r$, $a$ e $b$ hanno lo stesso MCD di $b$ e $r$.

[13] Ovviamente le radici quadrate dei numeri positivi si possono eseguire nell'ambito dei numeri reali, ma qui le nostre considerazioni sono limitate all'insieme dei numeri naturali.

Per stabilire la calcolabilità della radice quadrata sono disponibili diversi algoritmi[14]. Si può ad esempio nuovamente ricorrere alla scomposizione in fattori primi: un numero è un quadrato perfetto se e solo se, nella sua scomposizione in fattori primi, gli esponenti dei fattori sono tutti pari, e la radice quadrata si ottiene dividendo per 2 tutti gli esponenti ed eseguendo il prodotto. Ad esempio:

$$\sqrt{3\,969} = \sqrt{3^4 \times 7^2} = 3^2 \times 7 = 63$$

$$\sqrt{24\,200} = \sqrt{2^3 \times 5^2 \times 11^2} \quad \text{non esiste}$$

Questo modo di procedere non è particolarmente efficiente poiché, come si è già detto a proposito dell'MCD, si basa sulla scomposizione in fattori primi. Non interessa in questa sede la rapidità degli algoritmi[15], anzi ora descriviamo un algoritmo del tutto inefficiente, ma utile per qualche ulteriore importante considerazione.

Supponiamo di voler calcolare $\sqrt{49}$ cercando per successivi tentativi il numero il cui quadrato è 49.

Consideriamo la formula $y^2 = x$, attribuiamo a $x$ il valore 49 e a $y$ i numeri a partire da 0:

$y = 0, 1, 2, 3, 4,...$

Verifichiamo se sussistono le uguaglianze via via ottenute fino ad ottenerne una vera:

$0^2 = 49$ (falsa)
$1^2 = 49$ (falsa)
$2^2 = 49$ (falsa)
$3^2 = 49$ (falsa)
$4^2 = 49$ (falsa)
$5^2 = 49$ (falsa)
$6^2 = 49$ (falsa)
$7^2 = 49$ (vera)

Dato che siamo pervenuti a un'uguaglianza vera, arrestiamo il procedimento e concludiamo che $\sqrt{49} = 7$.

Se vogliamo calcolare $\sqrt{12}$ con lo stesso procedimento, otteniamo:

$0^2 = 12$ (falsa)
$1^2 = 12$ (falsa)
$2^2 = 12$ (falsa)
$3^2 = 12$ (falsa)
$4^2 = 12$ (falsa)
$5^2 = 12$ (falsa) ….....

[14] Molti anni fa, già nella scuola primaria si illustrava un algoritmo per il calcolo della radice quadrata, simile a quello con cui si esegue la divisione, oggi non più praticato data la sua macchinosità. Quando si devono calcolare radici quadrate, è ormai abituale ricorre a calcolatrici scientifiche. Si faccia una ricerca per stabilire quali sono gli algoritmi impiegati nelle calcolatrici (e nei calcolatori).

[15] Nella Teoria della computabilità si introducono "misure" per confrontare l'efficienza degli algoritmi. Questi aspetti più complessi vanno rimandati a studi successivi.

e non si ottiene mai un'uguaglianza vera, quindi $\sqrt{12}$ non esiste. È importante osservare che questo procedimento va avanti all'infinito senza arrestarsi, poiché i numeri che sostituiamo via via a *y* sono infiniti. Tuttavia, nonostante questa stranezza, il procedimento ha tutte le caratteristiche che abbiamo in precedenza attribuito agli algoritmi. Le istruzioni sono in numero finito e la radice quadrata, se esiste, viene ottenuta in un numero finito di passi. Nelle precedenti caratteristiche non abbiamo imposto alcuna condizione nel caso in cui la soluzione (in questo caso il valore della radice quadrata) non esiste, in particolare non si è richiesto che l'algoritmo esegua sempre un numero finito di passi.

Nell'esempio della radice quadrata si può facilmente evitare questo inconveniente. Infatti, la radice quadrata di un numero, se esiste, non è mai maggiore del numero stesso, e quindi, se l'uguaglianza $y^2 = x$ non risulta mai verificata fino a quando a *y* sostituiamo il valore di *x*, possiamo arrestare il procedimento e concludere che $\sqrt{x}$ non esiste[16].

[16] In realtà potremmo arrestare il procedimento già quando $y^2$ supera *x*. Gli aspetti di efficienza degli algoritmi, molto importanti nella pratica, non lo sono in questa sede, poiché il nostro obiettivo è illustrare, a grandi linee, alcuni sviluppi particolarmente significativi della teoria della computabilità, il cui approfondimento supera quanto si può introdurre nella scuola secondaria.

## DIAGRAMMI DI FLUSSO

Per illustrare meglio quanto esposto in precedenza utilizziamo la rappresentazione abituale degli algoritmi mediante i cosiddetti *diagrammi di flusso*[17].

Nella figura 1 è illustrato mediante un diagramma di flusso l'algoritmo euclideo delle divisioni successive, il quale calcola l'MCD di due numeri indicati con A e B:

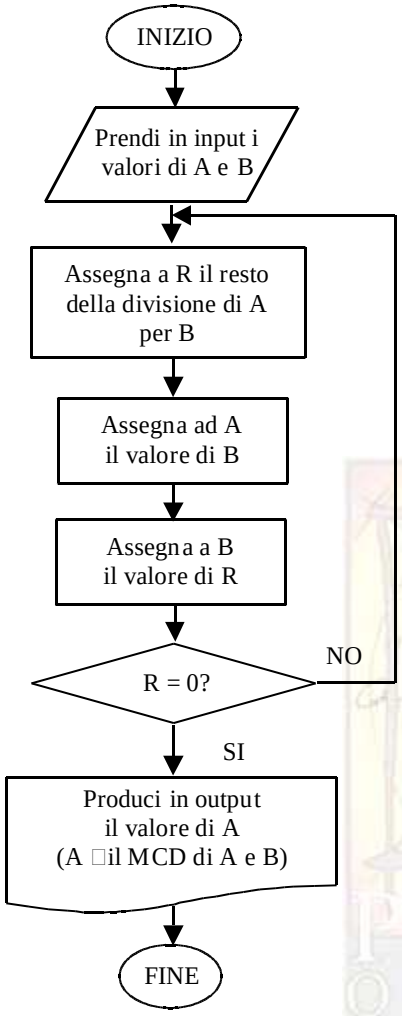


*Figura 1*

Nel rombo è inserita la condizione "R = 0?". Quando R = 0 (la risposta è "sì") il procedimento termina e si ottiene in output MCD(A, B). Se R ≠ 0 (la risposta è "no"), si esegue la successiva divisione, come abbiamo illustrato in precedenza. Nel diagramma di flusso è visualizzato con le frecce il ciclo che fa ripetere l'istruzione di eseguire una nuova divisione. Dato che prima o poi si ottiene R = 0, l'algoritmo termina sempre fornendo in output il massimo comun divisore cercato.

Se rappresentiamo il diagramma di flusso dell'algoritmo prima descritto per il calcolo della radice quadrata (figura 2), si vede che il ciclo cessa di essere ripetuto solo se *x* è un quadrato perfetto e in tal caso fornisce la radice quadrata *y* di *x*. Se *x* non è un quadrato perfetto, la con-

[17] I diagrammi di flusso sono frequentemente impiegati nella letteratura e in Internet si può trovare facilmente la loro descrizione. Dato che la loro lettura è agevole, non ci soffermiamo su di essi per non allungare troppo il nostro intervento.

dizione nel rombo non è mai verificata e l'algoritmo non termina. In questo secondo caso si dice talvolta che l'algoritmo va *in loop*[18]:

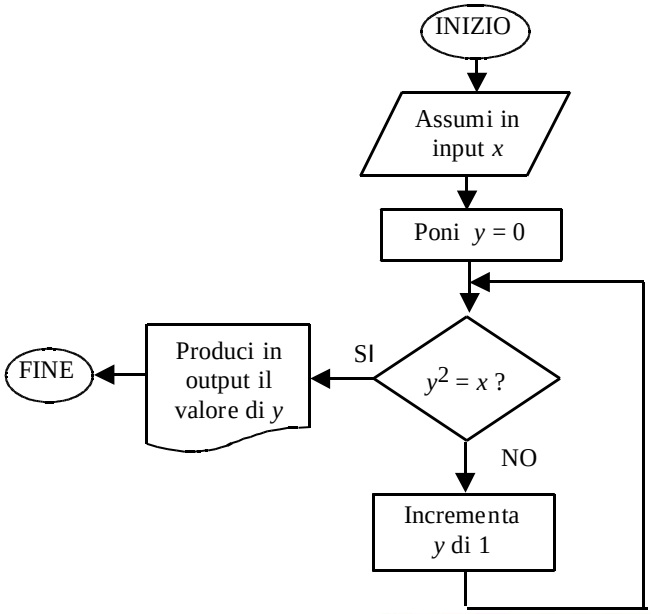


*Figura 2*

L'inconveniente viene eliminato se si modifica l'algoritmo come in figura 3, arrestando l'esecuzione del ciclo quando $y$ supera $x$, circostanza che prima o poi si verifica sempre:

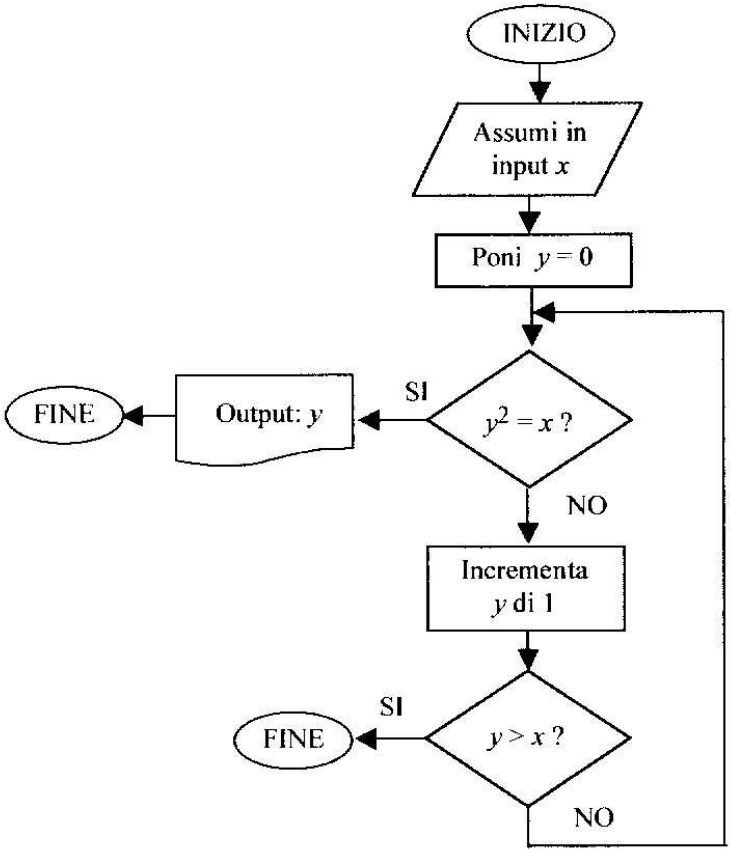


*Figura 3*

[18] A tutti è capitato che il calcolatore si sia "piantato" (sia andato *in loop*) e si sia dovuto procedere a uscite forzate da un programma o a dover riavviare il sistema operativo.

Il nuovo algoritmo termina sempre, o fornendo la radice quadrata di $x$ (se $x$ è un quadrato perfetto), o arrestandosi senza fornire output (volendo si può far produrre in output un messaggio del tipo "non esiste").

Negli algoritmi che si incontrano abitualmente nella pratica e in matematica si riesce sempre a modificare i *loop* che possono fare andare avanti all'infinito il procedimento, poiché, o il risultato esiste e lo si ottiene in un numero finito di passi oppure, quando il risultato non esiste, lo si può venire a sapere in un numero finito di passi.

> *Perché mai non abbiamo inserito tra le caratteristiche degli algoritmi la condizione apparentemente ovvia di dare sempre la risposta in un numero finito di passi* (*anche quando la soluzione non esiste*)?

È evidente come un algoritmo che possa andare avanti all'infinito appaia alquanto insoddisfacente. Infatti, un algoritmo ha la funzione di risolvere un problema dandoci una soluzione che spesso non sappiamo se esista o no. Se la soluzione non esiste e l'algoritmo va avanti all'infinito, non siamo in grado di stabilire la non esistenza della soluzione. Infatti, non possiamo sapere, mentre l'esecuzione dell'algoritmo è ancora in corso, se la soluzione ci sarà fornita in un tempo successivo (se esiste) oppure se non ci sarà mai fornita (perché non esiste). La risposta alla precedente domanda non è semplice, ma cercheremo di darla, almeno a livello generale, nel seguito (si veda la nota 24). Avremo così modo di accennare ad alcuni degli importanti sviluppi della teoria degli algoritmi, seppur uscendo da quanto può essere oggetto di studio nella scuola secondaria[19].

## FUNZIONI CARATTERISTICHE DI PROPRIETÀ E RELAZIONI

All'inizio dell'articolo abbiamo introdotto due concetti, la decidibilità delle proprietà e delle relazioni e la computabilità delle funzioni[20]. In realtà i due concetti si possono ricondurre l'uno all'altro mediante il concetto di *funzione caratteristica* di una proprietà o di una relazione. Illustriamo tale semplice concetto mediante due esempi. Consideriamo una proprietà in **N**, ad esempio "essere un quadrato perfetto". La seguente funzione $f$ di dominio **N** è detta funzione caratteristica della proprietà:

$$f(x)=\begin{cases}0 & \text{se } x \text{ è un quadrato perfetto}\\ 1 & \text{se } x \text{ non è un quadrato perfetto}\end{cases}$$

Analogamente, se consideriamo la relazione "essere divisore", la seguente funzione $g$ a due argomenti è la sua funzione caratteristica:

$$g(x,y)=\begin{cases}0 & \text{se } x \text{ è divisore di } y\\ 1 & \text{se } x \text{ non è divisore di } y\end{cases}$$

Le funzioni caratteristiche delle proprietà e delle relazioni, pertanto, sono funzioni aritmetiche che hanno come valori o 0 (se la proprietà o la relazione sussiste) o 1 (se la proprietà o la relazione non sussiste).

---

19 Gli argomenti trattati in questo articolo sono esposti in M. Frixione, D. Palladino, *La computabilità: algoritmi, logica, calcolatori*, Collana "Le Bussole", Carocci, Roma, 2011.

20 Come osservato nella nota 6, le operazioni sono funzioni, e quindi possiamo parlare di "funzioni" anziché di "funzioni e operazioni".

Vale la seguente proposizione:

> una proprietà o una relazione è decidibile se e solo se la sua funzione caratteristica è calcolabile.

Infatti, evidentemente, un algoritmo che stabilisce la decidibilità di una proprietà o di una relazione consente di calcolare la funzione caratteristica (basta modificare l'output "la proprietà o la relazione sussiste" in "0" e l'output "la proprietà o la relazione non sussiste" in "1"). Inversamente, un algoritmo che stabilisce la calcolabilità della funzione caratteristica consente di stabilire la decidibilità della proprietà o della relazione (basta invertire le modifiche degli output indicate nella precedente parentesi).

Queste considerazioni hanno lo scopo di consentire di affrontare uno solo fra i problemi della decidibilità e della computabilità (riconducendo uno dei due all'altro). Dato che le funzioni sono gli enti più importanti della matematica, d'ora in poi parleremo solo di "calcolabilità" delle funzioni.

Le funzioni che si incontrano abitualmente in aritmetica sono tutte calcolabili. La domanda che sorge spontanea è:

> *Esistono funzioni aritmetiche non calcolabili* ?

ossia funzioni tali che non esiste alcun algoritmo in grado di calcolarne i valori in corrispondenza degli argomenti per i quali tali valori esistono.

La risposta è affermativa e il modo di ottenerla costituisce uno degli argomenti più interessanti delle ricerche scientifiche del secolo scorso. Ad esso dedichiamo l'ultimo più impegnativo paragrafo.

## FUNZIONI ARITMETICHE NON CALCOLABILI

Un modo per dimostrare l'esistenza di funzioni aritmetiche non calcolabili si basa su considerazioni relative alla cardinalità degli insiemi infiniti. In matematica si sono definite due classi disgiunte di insiemi infiniti.

(a) Si dicono **numerabili** l'insieme **N** dei numeri naturali e gli insiemi che possono essere messi in corrispondenza biunivoca con **N**.

Ad esempio, sono numerabili l'insieme dei numeri pari, l'insieme dei numeri dispari, l'insieme dei numeri primi, l'insieme dei numeri quadrati perfetti, l'insieme dei numeri interi, l'insieme dei numeri razionali (ossia dei numeri decimali finiti o periodici).

(b) Si dicono **continui** l'insieme **R** dei numeri reali (insieme dei numeri decimali qualsiasi) e gli insiemi che possono essere messi in corrispondenza biunivoca con **R**.

Sono continui, oltre a **R**, l'insieme dei punti di un segmento, l'insieme dei punti di una retta, l'insieme dei punti del piano o dello spazio euclideo, l'insieme dei sottoinsiemi di **N**.

Si può dimostrare che gli insiemi numerabili sono "più piccoli" degli insiemi continui, nel senso che non si può instaurare alcuna corrispondenza biunivoca tra un insieme numerabile e uno continuo, ma solo una immersione iniettiva e non suriettiva del primo nel secondo[21]. Intui-

[21] Che si possa istituire un confronto fra gli insiemi infiniti e che l'infinità di **N** (la cardinalità degli insiemi numerabili) sia minore di quella di **R** (la cardinalità degli insiemi continui) è uno dei più

tivamente, ad esempio, un insieme numerabile di punti non può riempire completamente alcun segmento e ancor meno un'intera retta.

Ebbene, si può dimostrare che, essendolo l'insieme degli algoritmi, l'insieme delle funzioni aritmetiche calcolabili è *numerabile*[22] e che l'insieme delle funzioni aritmetiche è *continuo*. Ciò significa che vi sono più funzioni aritmetiche non calcolabili di quante ve ne siano di calcolabili. Anzi, dato che l'insieme delle funzioni aritmetiche non calcolabili è addirittura continuo, la calcolabilità può essere considerata una proprietà "rara" delle funzioni, in quanto posseduta da una quantità numerabile di esse. Questa conclusione può apparire sconcertante poiché, come si è detto, tutte le funzioni aritmetiche che si introducono nell'usuale sviluppo della matematica sono calcolabili. D'altra parte, è del tutto ovvio che l'attenzione si rivolga soprattutto a quelle funzioni i cui valori possono essere individuati mediante un algoritmo.

Il precedente ragionamento, che può essere condotto in modo rigoroso, ci ha consentito di concludere che esistono infinite funzioni non calcolabili, ma non ci fornisce nemmeno un solo esempio di una siffatta funzione. È opportuno che lo studente rifletta sul significato che l'esistenza ha in ambito matematico e come la si stabilisce. In primo luogo, l'esistenza è relativa all'ambito in cui si opera. Ad esempio, la radice quadrata di 2 non esiste in **N** poiché nessun numero naturale ha quadrato 2, ma esiste in **R**, che contiene molti più elementi di **N**. Inoltre, accade spesso che le dimostrazioni di esistenza siano ottenute per assurdo, senza esibire in modo esplicito qual è l'ente matematico che esiste. Questo è ad esempio il caso della dimostrazione precedente, che fa concludere che esistono funzioni non calcolabili senza esibirne nemmeno una[23].

*Osservazione importante*. Si noti che una funzione può essere calcolabile anche se non sappiamo calcolarne i valori. Infatti, affinché una funzione sia calcolabile, è sufficiente che *esista* un algoritmo che consente di individuarne i valori. Normalmente, l'esistenza dell'algoritmo è ottenuta esibendo l'algoritmo stesso (e dimostrandone la correttezza, ossia che ci fa ottenere i valori della funzione in corrispondenza degli argomenti per i quali il valore esiste). A volte però la dimostrazione di esistenza dell'algoritmo non consente di individuarlo. Un esempio chiarisce quanto affermato.

Vi sono varie proposizioni aritmetiche delle quali non sappiamo se sono vere o false. Una delle più celebri è la *congettura di Goldbach*: "Un qualsiasi numero pari maggiore di 2 è uguale alla somma di due numeri primi". Tale congettura è stata verificata per un'enorme quantità di numeri pari, e continua ad esserlo con l'impiego dei più potenti calcolatori. Ma, dato che i numeri pari sono infiniti, una verifica caso per caso può condurre solo a falsificare la congettura, se si pervenisse a individuare un numero pari che *non* è la somma di due numeri primi, cosa che fino ad oggi non si è realizzata. La verità della congettura, che contempla infiniti casi, può essere ottenuta solo mediante una dimostrazione, che finora nessuno è riuscito a trovare. Il termine "congettura" significa che il suo essere vera o falsa è un problema attualmente non ancora risolto.

---

importanti risultati della teoria degli insiemi e dell'intera matematica. Esso è stato ottenuto nel 1872 dal matematico tedesco Georg Cantor (1845-1918).

22 Le funzioni aritmetiche calcolabili sono sicuramente infinite, poiché sono calcolabili le infinite funzioni *costanti*, quelle che assumono lo stesso valore $k$ in corrispondenza di qualsiasi argomento $x$: $f(x) = k$ per ogni numero naturale $x$. L'algoritmo che stabilisce la loro calcolabilità è costituito dall'unica istruzione: "dai come output $k$ in corrispondenza di ogni input $x$".

23 Quello precedente, infatti, è un ragionamento per assurdo. Se tutte le funzioni aritmetiche, che formano un insieme continuo, fossero calcolabili, allora il loro insieme sarebbe numerabile (l'insieme delle funzioni calcolabili è numerabile). Ciò è assurdo poiché uno stesso insieme non può essere sia numerabile, sia continuo.

Consideriamo la funzione aritmetica $f$ tale che, per ogni $x$:

$$f(x)=\begin{cases}0 & \text{se la congettura di Goldbach è vera}\\ 1 & \text{se la congettura di Goldbach è falsa}\end{cases}$$

La funzione *f*, in base alla definizione, è calcolabile. Infatti, essa è una funzione costante (ha sempre valore 0 o ha sempre valore 1) e quindi *esiste* un algoritmo che ne calcola i valori. Però non sappiamo se è l'algoritmo che dà sempre output 0 o quello che dà sempre output 1, e quindi, fino a che la congettura di Goldbach resterà un problema aperto, non sapremo calcolare i valori di *f*.

La definizione di algoritmo fin qui utilizzata è sufficiente per gli scopi didattici dell'insegnamento secondario. Tuttavia, non è tale da consentire di esibire un esempio di funzione non calcolabile. Per poter dimostrare che una funzione *f* non è calcolabile, occorre provare che *non* esiste alcun algoritmo in grado di individuarne i valori in corrispondenza degli argomenti per i quali il valore esiste. Per riuscire in questa impresa è necessario poter considerare l'insieme di *tutti* gli algoritmi e dimostrare che ad esso non appartiene alcun algoritmo che calcola i valori di *f*.

A partire dagli anni trenta del secolo scorso, numerosi studiosi hanno proposto varie definizioni di algoritmo e di funzione calcolabile più precise di quella più intuitiva prima proposta. In sintesi, hanno individuato, seguendo ciascuno una propria strategia, un preciso insieme di funzioni aritmetiche che può essere identificato con l'insieme delle funzioni calcolabili[24]. Per rendere più chiaro il discorso, facciamo riferimento a una di queste caratterizzazioni, ossia quella ottenuta dal matematico inglese Alan Turing (1912-1954). Questi introdusse delle macchine astratte, oggi dette *macchine di Turing* MT, definite in modo preciso, in grado di eseguire il calcolo dei valori di funzioni aritmetiche impiegando istruzioni opportunamente formulate. Si può quindi definire in modo rigoroso l'insieme delle funzioni aritmetiche MT-computabili e ciò ha consentito di esibire delle funzioni che *non sono* MT-computabili e di stabilire che vi sono problemi *indecidibili*, ossia proprietà e relazioni la cui funzione caratteristica non è MT-computabile.

Riepilogando, Turing definì rigorosamente l'insieme delle funzioni MT-computabili. Dato che i procedimenti eseguiti da una MT sono sicuramente algoritmi nel senso della definizione prima adottata, vale la seguente implicazione:

*se una funzione f è MT-computabile, allora f è calcolabile*

L'implicazione inversa:

*se una funzione f è calcolabile, allora f è MT-computabile*

invece, è una proposizione che non si può dimostrare rigorosamente, dato che, come si è detto, la calcolabilità delle funzioni è un concetto che, basandosi sulla nozione di algoritmo prima esposta, non è precisata in modo tale da poter essere impiegata in una dimostrazione rigorosa. Essa viene detta *Tesi di Church*, dal nome del logico americano Alonzo Church (1903-1995)

[24] Nell'ambito di queste ricerche è emersa la necessità di rivolgere l'attenzione a funzioni parziali (che possono non avere valore per alcuni argomenti, come ad esempio la radice quadrata) e non ci si può limitare a considerare solo le funzioni definite per ogni argomento. Questa è la ragione principale per la quale si assumono come algoritmi anche procedimenti che possono andare avanti all'infinito. Per l'illustrazione di questa non facile tematica rinviamo al testo citato nella nota 19, un cui capitolo è tra l'altro dedicato alle MT. Materiale relativo alle macchine di Turing si può comunque reperire facilmente in Internet.

che è stato il primo a formularla, quasi contemporaneamente a Turing, nel 1936. Tale tesi è stata confermata in tutte le ricerche successive, nel senso che, nonostante i molteplici tentativi, nessuno è finora riuscito ad individuare una funzione calcolabile che non sia MT-computabile. La Tesi di Church, pertanto, è accettata nel mondo scientifico come un vero e proprio principio fisico, il quale è impiegato per ottenere risultati di indecidibilità e di non calcolabilità[25].

Accettando la Tesi di Church si sono individuati vari problemi indecidibili e diverse funzioni non calcolabili. Ciò che rende importanti questi risultati, al di là del loro particolare interesse intrinseco, è che tutti i calcolatori realizzati a partire dalla metà del secolo scorso non sono in grado di svolgere compiti che una MT non sia in grado di svolgere. Vi sono quindi problemi che nessun calcolatore con l'architettura di quelli attualmente esistenti potrà mai risolvere: i limiti dei calcolatori, che è bene siano sempre tenuti presenti, sono stati individuati decenni prima della loro realizzazione!

*Alan Mathison Turing (1912-1954)*

*Alonzo Church (1903-1995)*

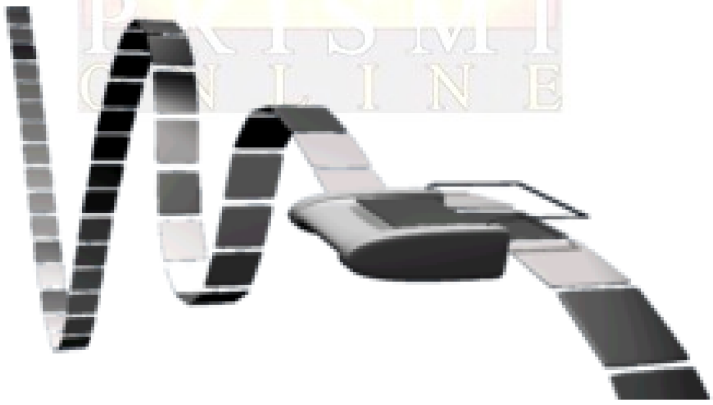

*Immagine di una Macchina di Turing*

[25] Un problema indecidibile individuato da Turing fu il *problema dell'arresto delle MT*: non esiste alcun algoritmo il quale, assunti in input una qualsiasi MT e un suo argomento, calcoli se la MT termini o meno la sua computazione in un numero finito di passi. Al problema dell'arresto si collegano altri problemi molto importanti sia nella pratica, sia nell'indagine delle teorie matematiche. Le ricerche cui si è accennato, così importanti per l'informatica, hanno avuto origine nell'ambito della logica matematica e dello studio dei fondamenti della matematica dell'inizio del secolo scorso.