LA MATEMATICA NEGLI ALGORITMI

Fabrizio Luccio, Pisa 2016

La decidibilità

il concetto di algoritmo

non esistono dimostrazioni gratis

La crescita esponenziale

rappresentazione e comunicazione

la complessità di calcolo

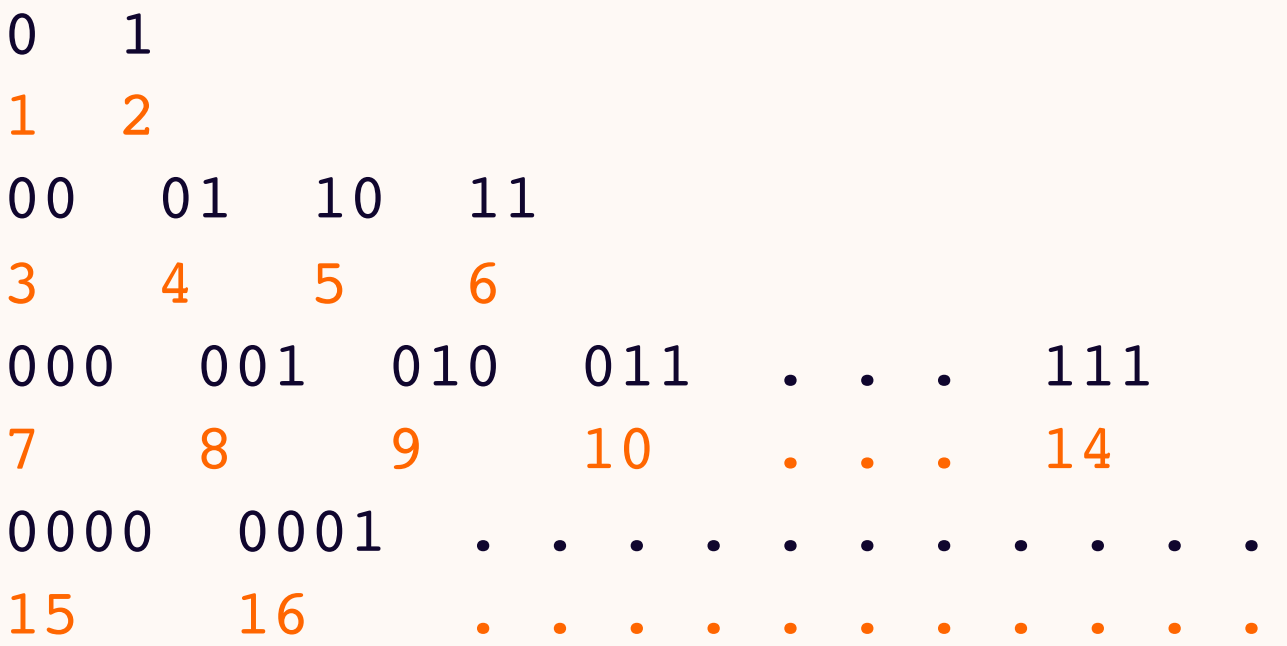
La casualità

la compressione dei dati

un aiuto dal caso

La decidibilità

nasce alla fine del 1800 dalla teoria degli insiemi infiniti che comporta il concetto di numerabilità



Le sequenze sono numerabili

	0	1	2	3	4	5	6	.	.
F0	1	1	0	1	0	1	1	.	.
F1	0	1	0	0	0	1	0	.	.
F2	0	0	0	1	1	0	1	.	.
F3	1	0	1	1	1	0	1	.	.
..	.	.	.	.	.	.	.	.	.

Tabella delle funzioni Fi: N ➔ {0,1}

	0	1	2	3	4	5	6	.	.
F0	1	1	0	1	0	1	1	.	.
F1	0	1	0	0	0	1	0	.	.
F2	0	0	0	1	1	0	1	.	.
F3	1	0	1	1	1	0	1	.	.
..	.	.	.	.	.	.	.	.	.

F3(4) = 1

	0	1	2	3	4	5	6	.	.
F0	1	1	0	1	0	1	1	.	.
F1	0	1	0	0	0	1	0	.	.
F2	0	0	0	1	1	0	1	.	.
F3	1	0	1	1	1	0	1	.	.
..	.	.	.	.	.	.	.	.	.

Fj(i)=0 se e solo se Fi(i)=1

Fj non appartiene all'elenco

Le funzioni non sono numerabili

La numerabilità degli algoritmi (sequenze) e la non numerabilità delle funzioni segna la nascita, all'inizio del 1900, della teoria della calcolabilità

e richiede di porre una definizione formale al concetto di algoritmo

Nel 1936 Alan Turing definisce l'algoritmo attraverso una "macchina" astratta e dimostra che il problema della terminazione è algoritmicamente indecidibile

Non esiste un algoritmo HALT che decide se un altro algoritmo arbitrario A, operando su dati arbitrari D, termina o no:

```
HALT(A,D)=true,  se A(D) termina

HALT(A,D)=false, se A(D) non termina
```

Il meccanismo di dimostrazione prende spunto dall'epistola di San Paolo a Tito . . . .

Per questa ragione ti ho lasciato a Creta:
perché tu metta ordine nelle cose . . . . .

Infatti vi sono molti ribelli, ciarloni e
seduttori delle menti, specialmente tra
quelli della circoncisione . . . . .

Uno dei loro, proprio un loro profeta, disse:
« I cretesi sono sempre bugiardi, male
bestie, ventri pigri ».

Consideriamo un nuovo algoritmo P che lavora su una sequenza A che rappresenta un algoritmo:

```
P(A)
while (HALT(A,A)=true} nulla
                else return ciao
```

P(P) termina se e solo se P(P) non termina !!!

È un'antinomia: il punto debole
è l'ammissione che HALT esista

$a^n + b^n = c^n$

non ha soluzione intera positiva per $n > 2$

è una famosa affermazione di Fermat
dimostrata da Andrew Wiles nel 1995
in più di 130 pagine

```
FERMAT
    for (i=6 on, i++)
      costruisci le quaterne q_i={n,a,b,c}
      con n>2, a,b,c>0, n+a+b+c=i;
      per ogni q_i:
         if (a^n+b^n=c^n) return.
```

FERMAT termina se e solo se l'affermazione di Fermat è falsa per una certa quaterna

Immaginiamo di avere un algoritmo HALT:

```
HALT(FERMAT)=true
      ⇒ l'affermazione è falsa

HALT(FERMAT)=false
      ⇒ l'affermazione è vera
```

Se esistesse (e io conoscessi) l'algoritmo HALT, esisterebbe un modo banale di dimostrare gratis qualsiasi congettura sui numeri interi

La crescita esponenziale

Impiegando un alfabeto di k simboli il numero p di "parole" di lunghezza n cresce esponenzialmente con n:

$$p = k^n$$

Parole: aaa aab aac . . . . zzz

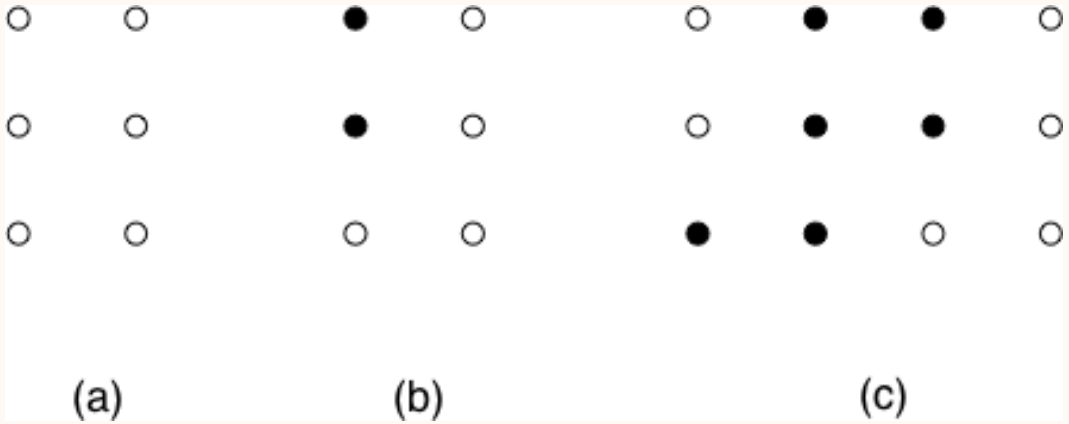
$k = 26,\ n = 3,\ p = 26^3 = 17.576$

per $k \geq 2$ si ha: $k^n > k^{n-1} + k^{n-2} + \ldots + k^1$

La crescita esponenziale è la legge matematica che ci permette di comunicare se si usa un alfabeto di almeno due caratteri!

Nell'informatica (e non solo) si sceglie un alfabeto binario: $p = 2n$

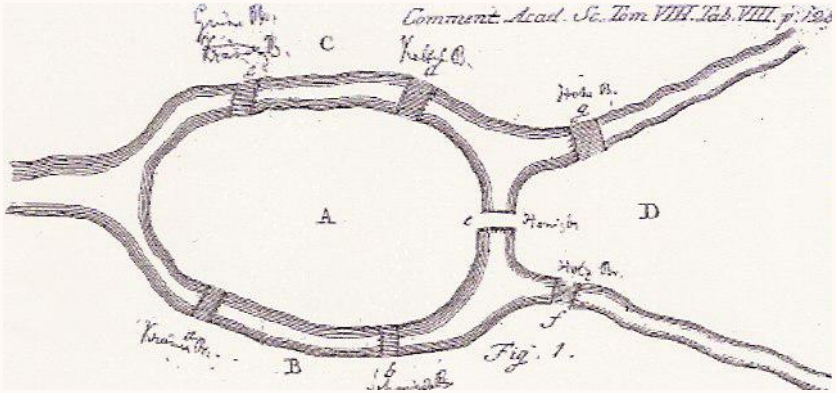
Il Braille: un codice binario sorprendente

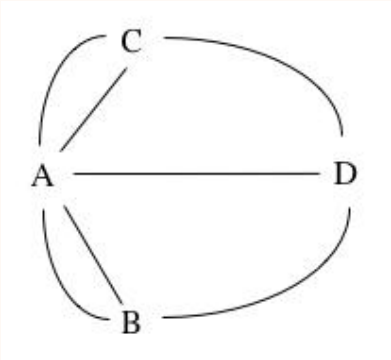


(a) Le sei posizioni dei punti
(b) La lettera B: i punti in rilievo sono in nero.
(c) il numero 2, composto da un "segno numeri" per il "cambio di ambiente", seguito dalla lettera B.

La complessità di calcolo

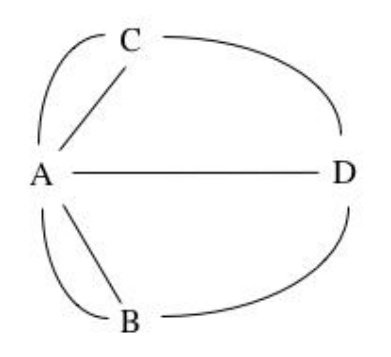
San Pietroburgo, 1735

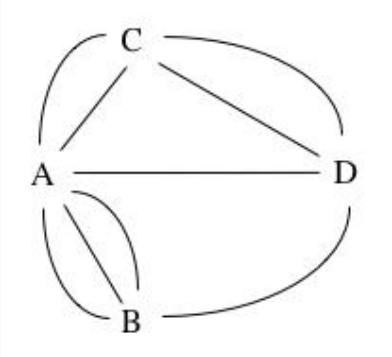




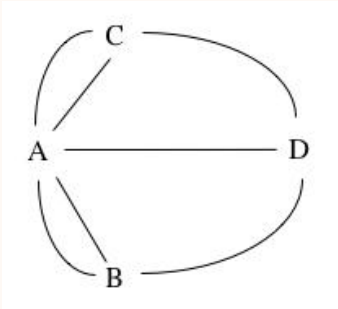
La condizione di Eulero:

Esiste un ciclo che traversa tutti gli archi (ponti) esattamente una volta se e solo se tutti i nodi (zone della città) hanno grado pari.

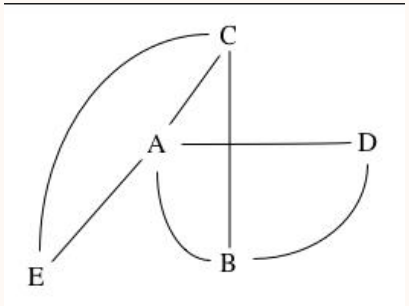




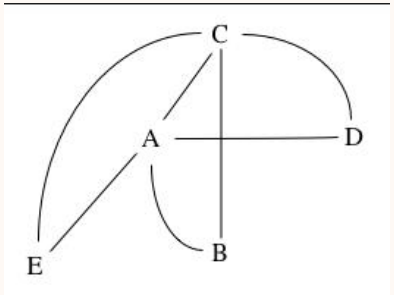
Il ciclo Hamiltoniano traversa tutti i nodi esattamente una volta



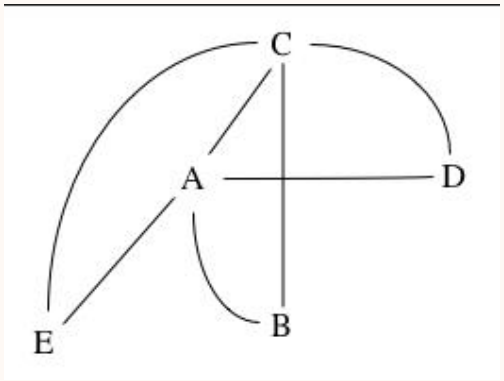
A B D C A



A D B C E A



? ?



A B C D E ?
A B C E D ?
. . . . . .

In linea di principio si devono fare n! = 120 prove

n =	2	3	4	5	6	7	8	9	10
n! =	2	6	24	120	720	5040	40320	362880	> 3.5M

n! cresce come $(n/e)^n$

La complessità di calcolo

è il "tempo" (cioè il numero di operazioni elementari) necessario per risolvere un problema.

Sono complessi i problemi che richiedono tempo esponenziale

Complessità polinomiale e esponenziale

Un algoritmo di complessità cn^s risolve un problema P su n dati in tempo t su un computer A. Lo stesso algoritmo, su un computer k volte "più veloce" di A, risolve in tempo t lo stesso problema su m dati:

$$cn^s = t \qquad cm^s = kt \qquad \text{dunque} \qquad m = k^{1/s} n$$

Ripetiamo il ragionamento con un algoritmo di complessità $c2^n$:

$$c2^n = t \qquad c2^m = kt \qquad \text{dunque} \qquad m = n + \log_2 k$$

Le due principali classi di complessità

P è la classe dei problemi che si risolvono in tempo polinomiale.

NP è la classe dei problemi che si verificano in tempo polinomiale (ma si sanno risolvere solo in tempo esponenziale).

P = NP ?

Un interessante esempio di complessità

Le equazioni diofantee

$ax + by + c = 0$ è in P

linea retta: risolubile in tempo polinomiale nella lunghezza della rappresentazione di a, b, c

$ax^2 + by + c = 0$ è in NP

parabola: risolubile in tempo polinomiale nel valore di a, b, c

$E = 0$

equazione di grado arbitrario e qualunque numero di variabili: non è risolubile ! algoritmicamente

La distinzione tra problemi polinomiali e esponenziali è alla base della crittografia . . . .

che utilizza funzioni "one way"
cioè "facili da calcolare" e "difficili"
da invertire

Un algoritmo crittografico non può essere mantenuto segreto a lungo . . . .

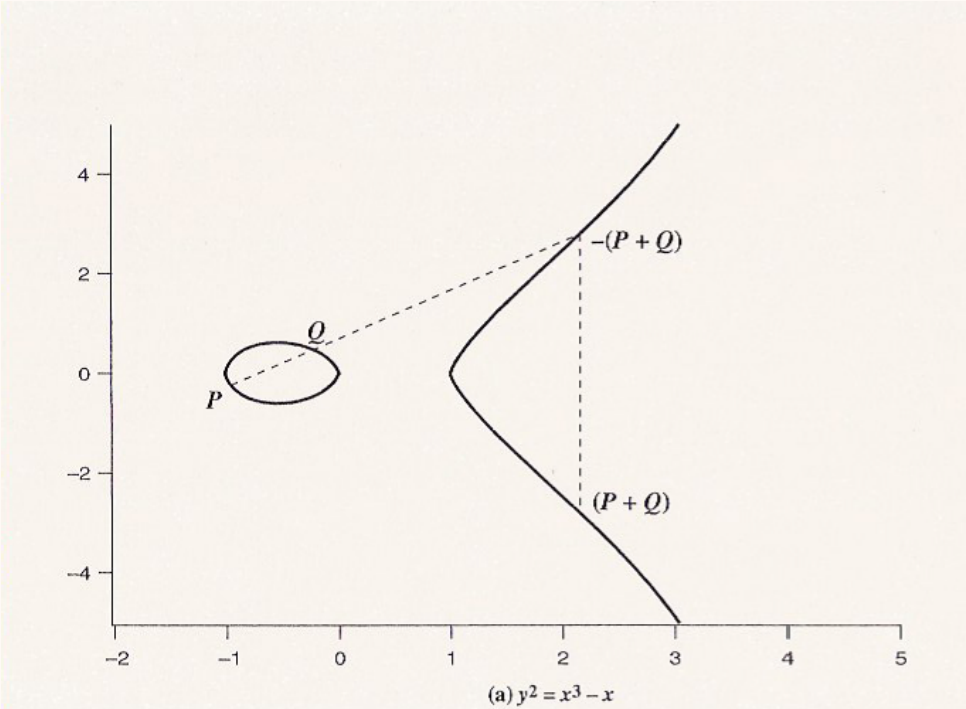
quindi deve essere pubblico, ma basato su una chiave segreta e possibilmente casuale

Vediamo due partner possano costruire una loro chiave segreta con un metodo noto a tutti, mentre tutti ne intercettano la comunicazione

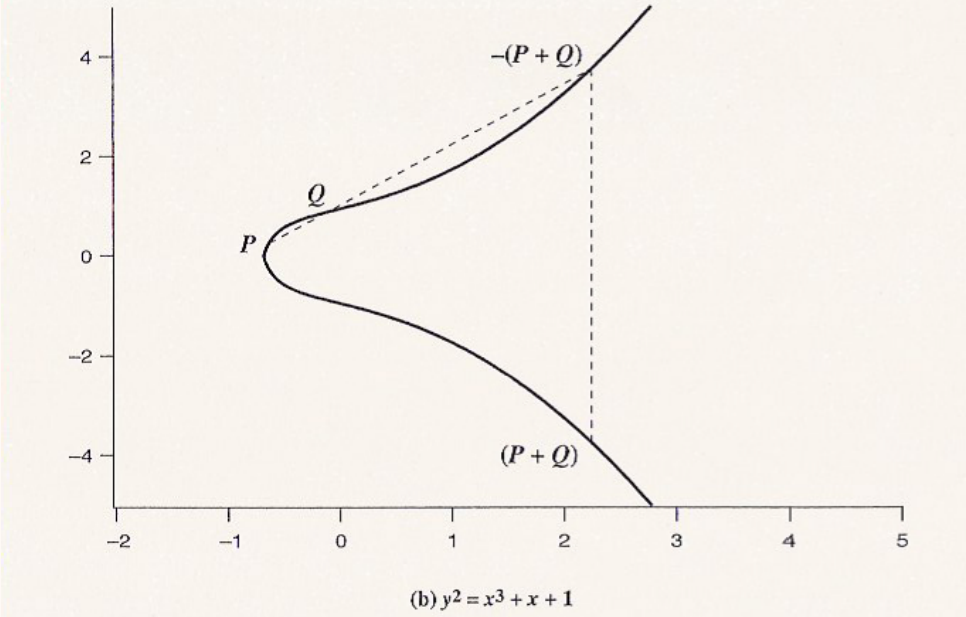
Le curve ellittiche

$y^2 = x^3 + ax + b$

Si considerano solo i punti a coordinate intere (e si opera "in modulo")



(a) $y^2 = x^3 - x$



(b) $y^2 = x^3 + x + 1$

Un punto P può essere sommato a sé stesso mediante la tangente in P alla curva

Moltiplicazione di P per una un numero intero k:

$$P + P + \ldots + P = k\,P$$

La moltiplicazione è eseguita con raddoppi e addizioni. Per esempio se $k = 13$:

$$13\,P = P + 4P + 8P = P + (2(2P)) + (2(4P))$$

Per $R = k P$

dati k e P, si calcola R in tempo "breve"

dati R e P, si sa calcolare k solo in tempo esponenziale

Costruzione di una chiave tra Alice e Bob

- I due concordano su una curva da usare e su un suo punto P (che possono essere noti a tutti)
- Alice sceglie a caso un intero segreto α, calcola αP e lo invia a Bob
- Bob sceglie a caso un intero segreto β, calcola βP e lo invia ad Alice
- Alice calcola la chiave comune $k = \alpha\beta P$
  Bob calcola la chiave comune $k = \beta\alpha P$

La casualità

Esù
religione Ifé

Pierre-Simon Laplace

Essai philosophique sur les probabilités (1814)

```
C C C C C C C C C C C C C C C C C C C C
G E N E R A - 2 0 - C

C P P C C C P C P C P C C P P P C C P C
G E N E R A " C P P C . . . . . . . . . C "
```

Una sequenza è casuale se non ammette alcun algoritmo di generazione (o regola per descriverlo) più corto della sequenza stessa

Teoria della casualità algoritmica

la complessità algoritmica $K(S)$ di una sequenza S è la lunghezza del più corto "programma" che la genera in un sistema universale di calcolo

una sequenza S è casuale se $K(S) \geq |S|$

La *casualità* è definita come proprietà intrinseca di una sequenza *indipendentemente dalla sorgente che l'ha generata*. Una sequenza che risponde a una semplice regola è *non casuale* anche se è stata generata perfettamente a caso.

Questa definizione di casualià *prescinde dall'esistenza*, filosoficamente discutibile, di sorgenti che generano elementi a caso.

Notiamo ora che, per il fenomeno della crescita esponenziale, le sequenze lunghe n sono più numerose di quelle lunghe meno di n indipendentemente dall'alfabeto impiegato . . . .

dunque le sequenze casuali esistono

ma stabilire se un'arbitraria sequenza è casuale è un problema indecidibile

Jorge Luis Borges
Del rigor en la ciencia (1946)

En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el Mapa de una sola Provincia ocupaba toda una Ciudad, y el Mapa del Imperio, toda una Provincia. Con el tiempo, estos Mapas Desmesurados no satisficieron y los Colegios de Cartógrafos levantaron un Mapa del Imperio, que tenía el Tamaño del Imperio y coincidía puntualmente con él.

La compressione dei dati

Fissato un sistema di riferimento, la massima compressione di una sequenza S si ottiene sostituendo S con l'algoritmo più corto che lo genera.

Quindi le sequenze casuali, e solo esse, non possono essere compresse.