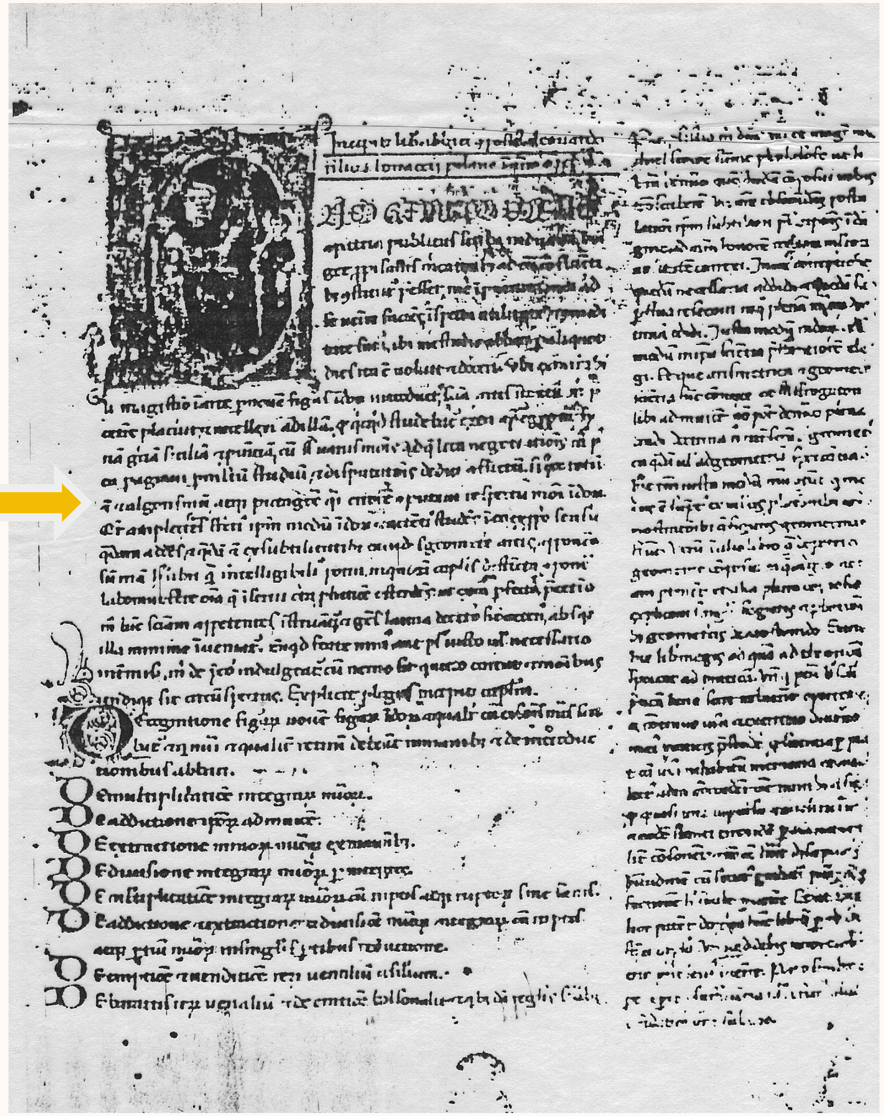
La matematica negli algoritmi

Maat: dea egizia dell'ordine

Liber Abaci
1202



Tre concetti di base

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
- algoritmi randomizzati

La decidibilità

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

0 1

00 01 10 11

000 001 010 011 . . . 111

0000 0001 1111

00000 . . . .

. . . .

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	.	.
..	.	.	.	.	.	.	.	.	.

Fj(i) = NOT Fi(i) ???

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
```

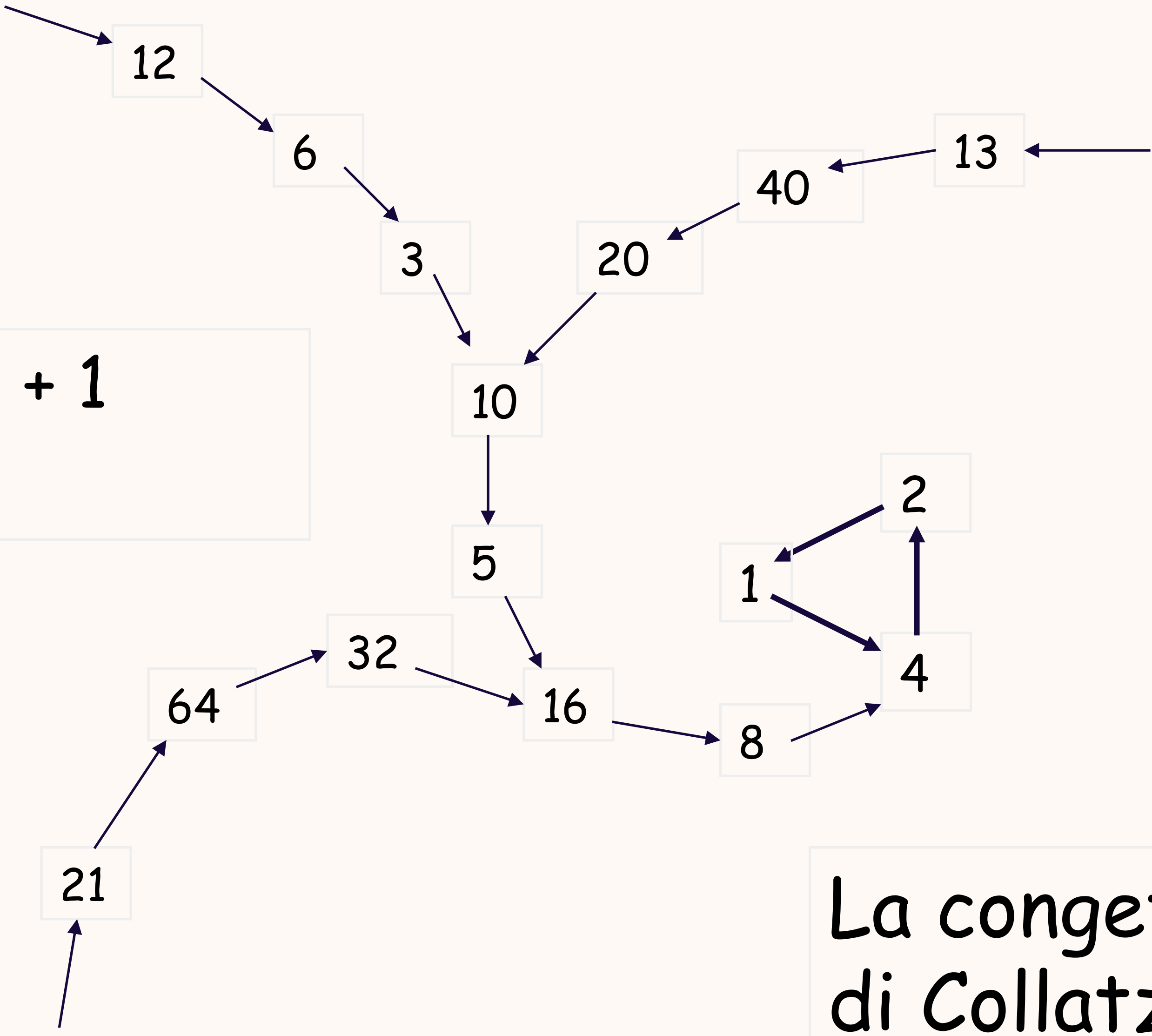
Il meccanismo di dimostrazione prende spunto da un'epistola di San Paolo . . . .

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 !!!

$3n + 1$



La congettura di Collatz 1937

```
CICLO(n)
   repeat
     if (n è pari) n=n/2 else n=3n+1
   until n=1
```

$n = 1$

$n = 2 , 1$

$n = 3 , 10 , 5 , 16 , 8 , 4 , 2 , 1$

. . . . . .

CICLO(*n*) termina se e solo se la congettura di Collatz è vera per l'intero *n*

Immaginiamo di avere un algoritmo HALT:

`HALT(CICLO(`n`))=true`

$\Rightarrow$ `la congettura è vera per` n

`HALT(CICLO(`n`))=false`

$\Rightarrow$ `la congettura è falsa per` n

```
COLLATZ
   for (n = 1 to ∞)
      if (HALT(CICLO(n) = false) stop
```

se la congettura di Collatz è vera (per ogni intero) l'algoritmo COLLATZ non tremina; altrimenti termina sul primo valore *n* che rende falsa la congettura

Avremmo: HALT(COLLATZ) = false

Se esistesse (e io conoscessi) l’algoritmo HALT, avrei 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: $k^n > k^{n-1} + k^{n-2} + \ldots + k^1$

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

N. B.

~~Sufficerent in~~
Sufficerent in hac Arithmeticâ duo characteres, nempe a et o. et hoc ultimo semper utemur pro zero, ut observetur ~~…~~ in exprimendis numeris uniformitas.

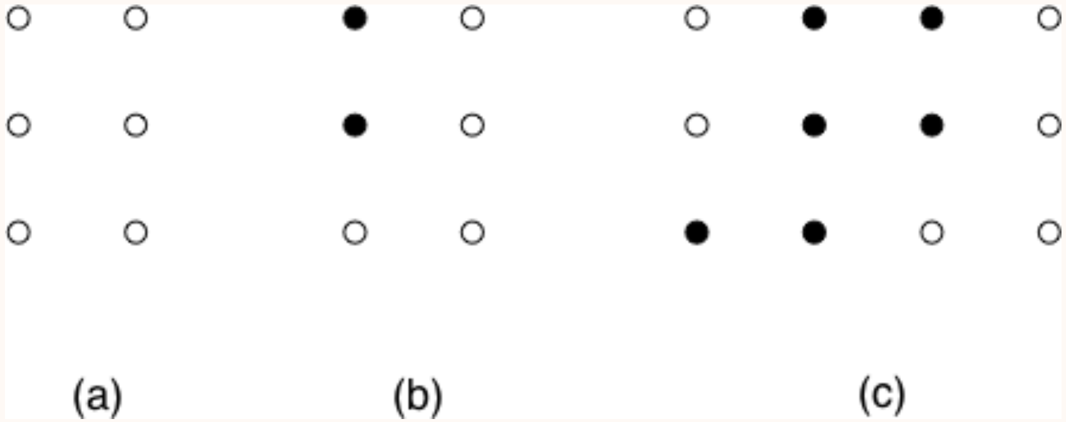
o	o
a	1
ao	2
aa	3
aoo	4
aoa	5
aao	6
aaa	7
aooo	8
aooa	9
aoao	10
aoaa	11
aaoo	12
aaoa	13
aaao	14
aaaa	15
aoooo	16

aoooo	16
aoooa	17
aooao	18
aooaa	19
aoaoo	20
aoaoa	21
aoaao	22
aoaaa	23
aaooo	24
aaooa	25
aaoao	26
aaoaa	27
aaaoo	28
aaaoa	29
aaaao	30
aaaaa	31
aooooo	32. &c.

Meditatio Proemialis.

Da: Johannes Caramuel: *Matesis Biceps Vetus et Nova*, 1670

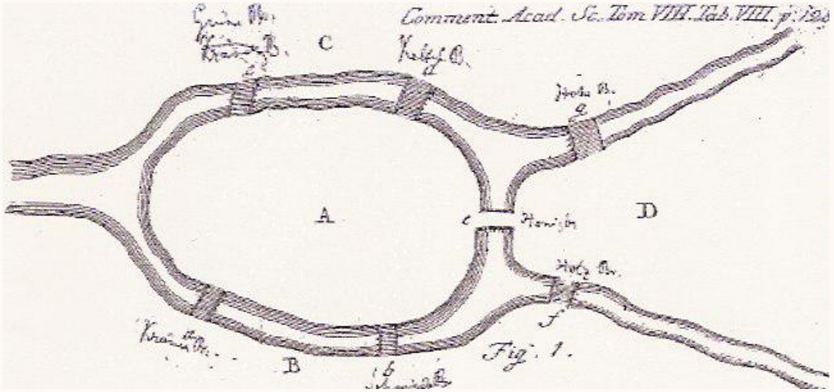
Il Braille: un codice binario sorprendente

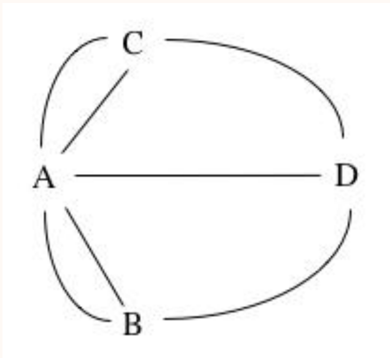


(a) Le sei posizioni dei punti
(b) La lettera B: i punti in rilievo sono in nero.
(c) il numero 2, composto da un "segno numeri" per il "cambio di ambiente", seguito dalla lettera B.

La complessità di calcolo

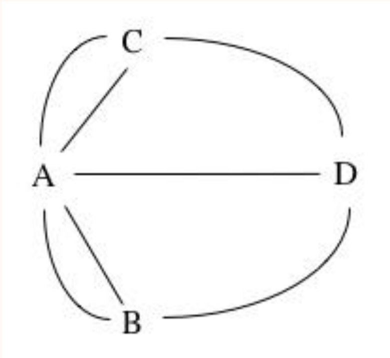
San Pietroburgo, 1735

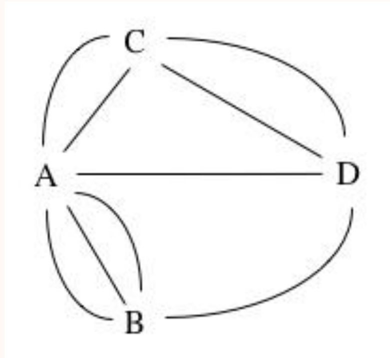




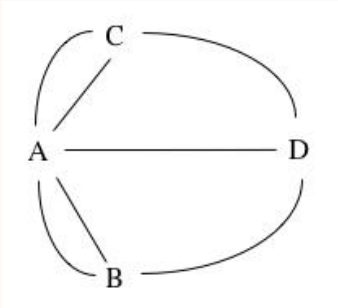
La condizione di Eulero:

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.

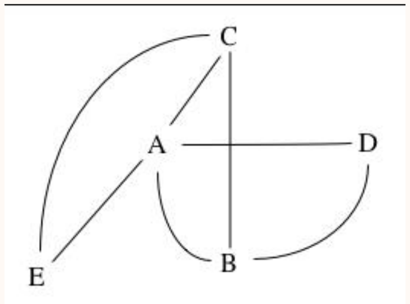




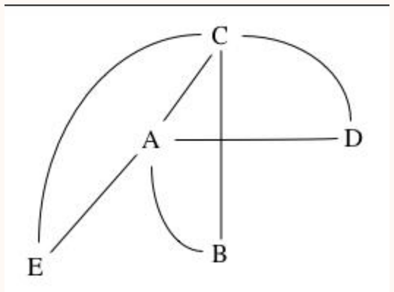
Il ciclo Hamiltoniano traversa tutti i nodi esattamente una volta



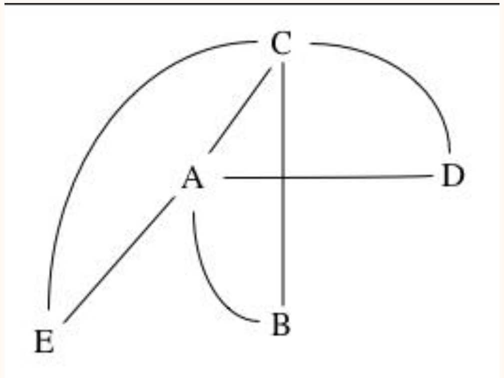
A B D C A



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

2	3	4	5	6	7	8	9	10
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 n^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 tempo t lo stesso problema su m dati:

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

Ripetiamo il ragionamento con un algoritmo di complessità 2^n:

$$2^n = t \qquad 2^m = k\,t \qquad \text{dunque} \qquad m = n + \log_2 k$$

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 ?

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 . . . .

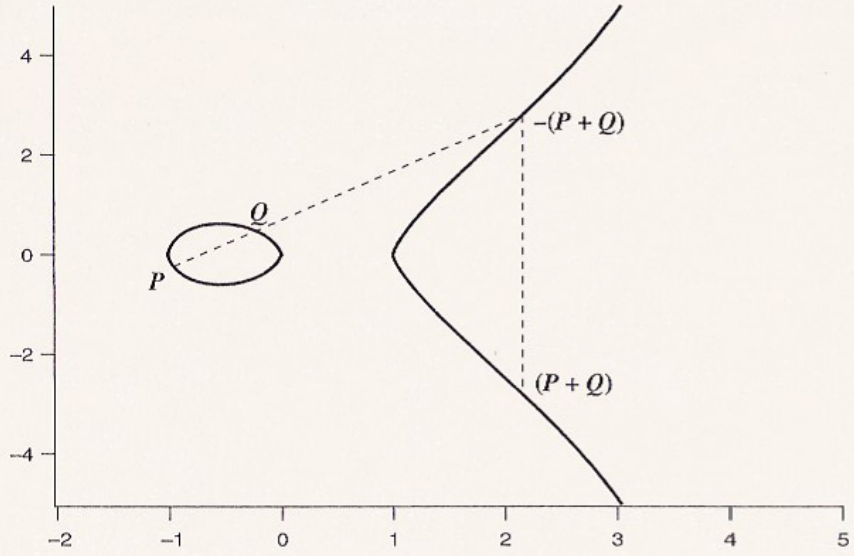
quindi deve essere pubblico, ma basato su una chiave segreta e possibilmente casuale

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

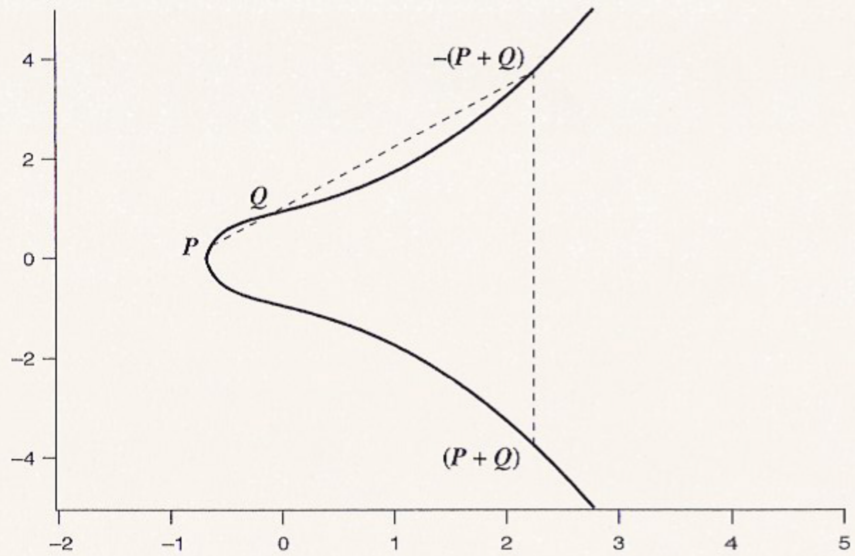
Le curve ellittiche

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

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 $k = \beta\alpha P$