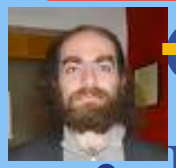
Il problema di P e NP

Calcolo efficiente, sicurezza di Internet,
e i limiti della conoscenza umana

Linda Pagli
Dipartimento di Informatica
Università di Pisa

Clay Math Institute
Problemi del Millennio da $1M ciascuno

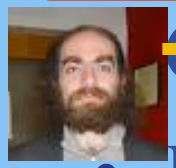
- Congettura di Birch and Swinnerton-Dyer
- Congettura di Hodge
- Equazioni di Navier-Stokes
- P vs NP
- ~~Congettura di Poincaré~~



- Ipotesi di Riemann
- Teoria di Yang-Mills

Clay Math Institute
Problemi del Millennio da $1M ciascuno

- Congettura di Birch and Swinnerton-Dyer
- Congettura di Hodge
- Equazioni di Navier-Stokes
- P vs NP
- ~~Congettura di Poincaré~~
- Ipotesi di Riemann
- Teoria di Yang-Mills



Il più semplice da spiegare

Il più recente 1971

Introduzione

I computer sono velocissimi.

Certi problemi richiedono tantissimo tempo!

Cominciamo con un esempio molto semplice…

Un semplice esempio

$7 \times 13 = ?$

Problema della Moltiplicazione
(Risposta 91)

Un altro semplice esempio

$? \times ? = 91$

"Problema della fattorizzazione"

(Risposta: 7×13)

Un esempio più grande di moltiplicazione

1.634.733.645.809.253.848
443.133.883.865.090.859.
841.783.670.033.092.312.
181.110.842.389.333.100.
104.508.151.212.118.167.
511.579

X

1.900.871.281.664.822.113.
126.851.573.935.413.975
471.896.789.968.515.493.
666.638.539.088.027.103.
802.104.498.957.191.261.
465.571

= ?

La risposta è:

3.107.418.240.490.043.721.350.750.035.888.567.930.037.346.022.842.727.
545.720.161.948.823.206.440.518.081.504.556.346.829.671.723.286.782.
437.916.272.838.033.415.471.073.108.501.919.548.529.007.337.724.822.
783.525.742.386.454.014.691.736.602.477.652.346.609

Ha richiesto meno di un secondo di tempo di calcolo

Un esempio più grande di fattorizzazione

? x ? = 3.107.418.240.490.043.721.350.750.035.888.567.930.037.
346.022.842.727.545.720.161.948.823.206.440.518.081.
504.556.346.829.671.723.286.782.437.916.272.838.033.
415.471.073.108.501.919.548.529.007.337.724.822.
783.525.742.386.454.014.691.736.602.477.652.346.609

La risposta è:

1.634.733.645.809.253.848
443.133.883.865.090.859.
841.783.670.033.092.312.
181.110.842.389.333.100.
104.508.151.212.118.167.
511.579

x

1.900.871.281.664.822.113.
126.851.573.935.413.975
471.896.789.968.515.493.
666.638.539.088.027.103.
802.104.498.957.191.261.
465.571

Ha richiesto 20 anni di tempo di calcolo

Non ancora fattorizzato:

7403756347956171282804679609742957314259318889231289084
9362326389727650340282662768919964196251178439958943305021275853701189680982867331732731089309005525051168770632990723963807867100860969625379346505637963 59

Numero di 212 cifre: RSA-704

Erano stati offerti 30.000 $, ma la gara è stata chiusa nel 2007 e nessuno ha vinto il premio.

La fattorizzazione è un ingrediente della crittografia

Per aprire un lucchetto a combinazione occorrono pochi secondi conoscendo le 4 cifre della combinazione (PIN)

Se non si ha il PIN di 4 cifre occorrono 10.000 prove per scoprire la combinazione giusta

presi due numeri primi p, q

Calcolare $n = p \times q$ " è facile "

Calcolare p, q da n " è difficile "

perché si devono provare tutti i divisori di n che, come per il lucchetto, sono in numero esponenziale rispetto al numero di cifre di n

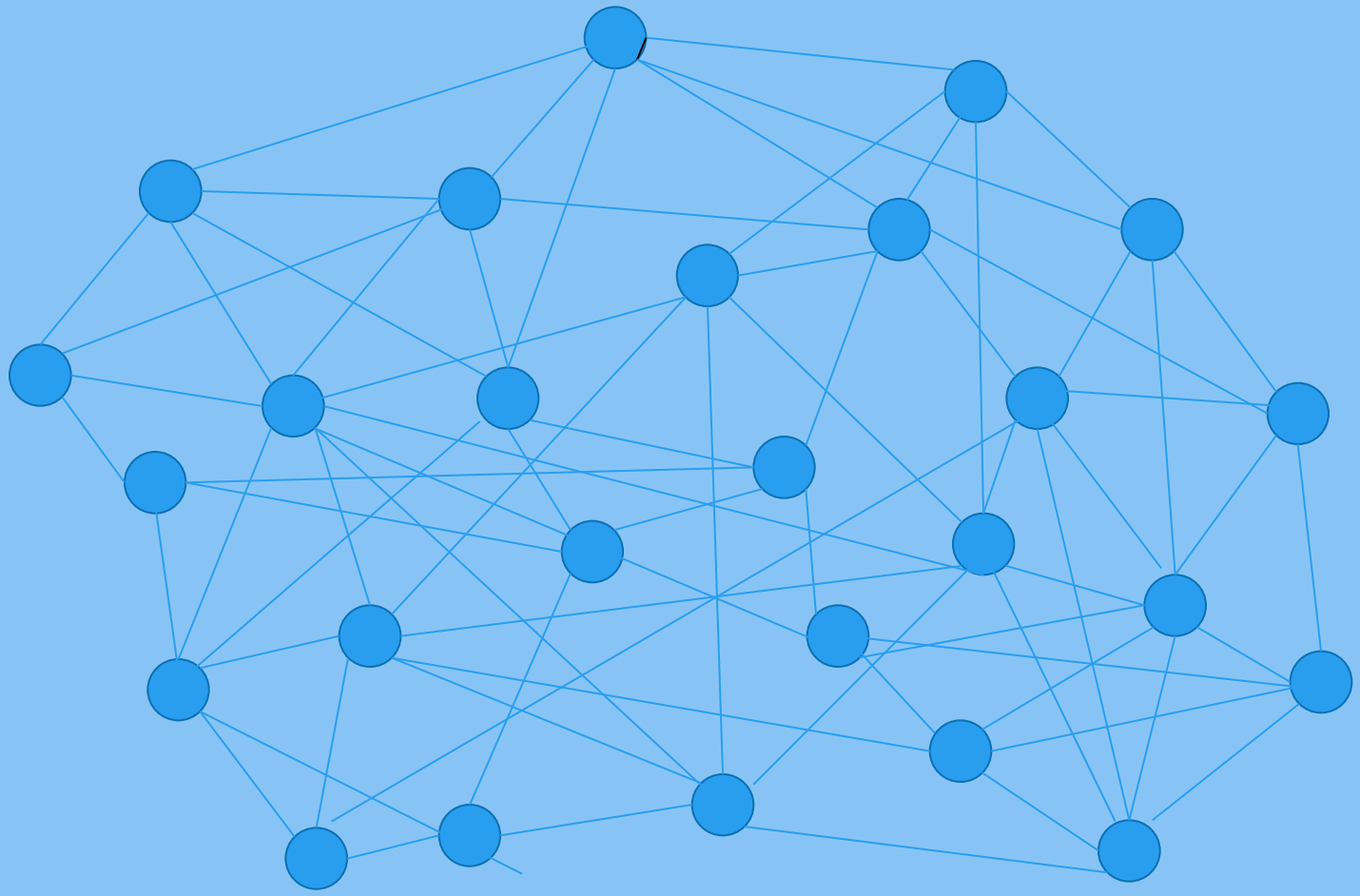
Ricerca col metodo “brute force” .

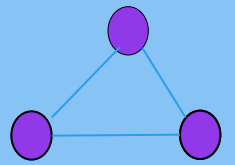
Ricerca esaustiva : molto lenta se lo spazio di ricerca è ampio.

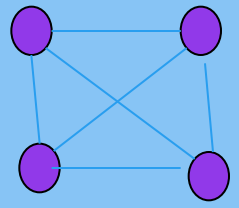
Ma la ricerca esaustiva è necessaria?

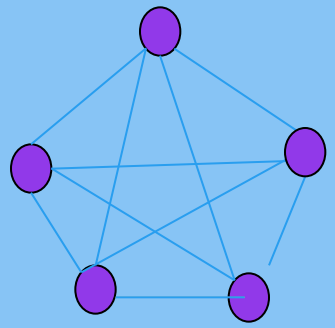
Non sappiamo rispondere

Problema della CLIQUE

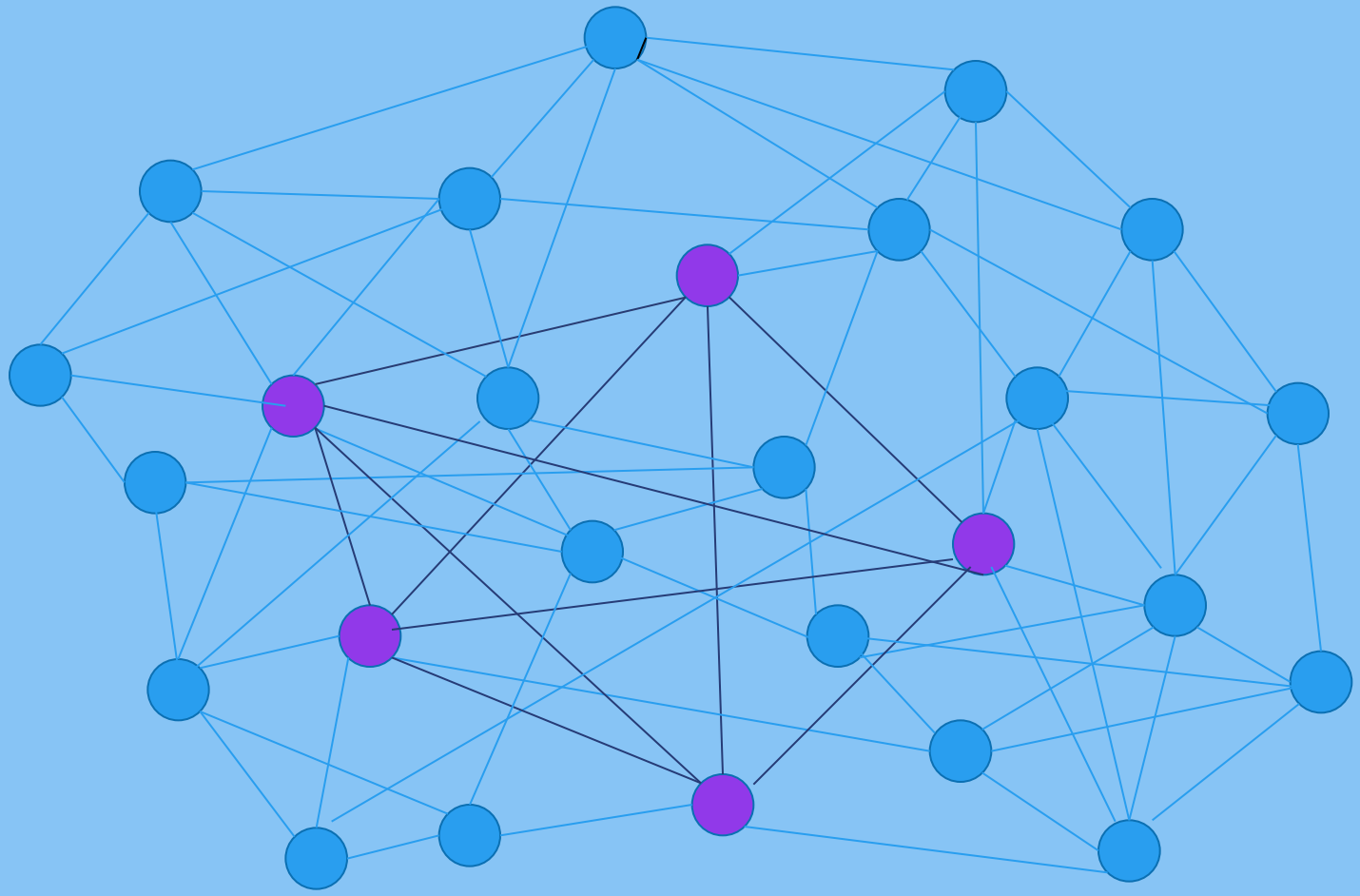


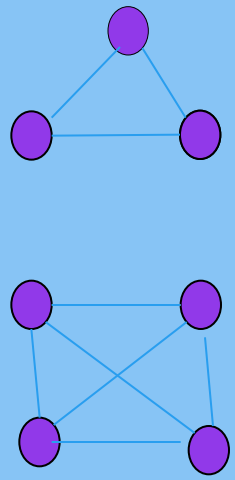


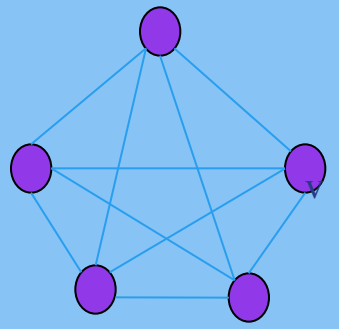




Problema della CLIQUE

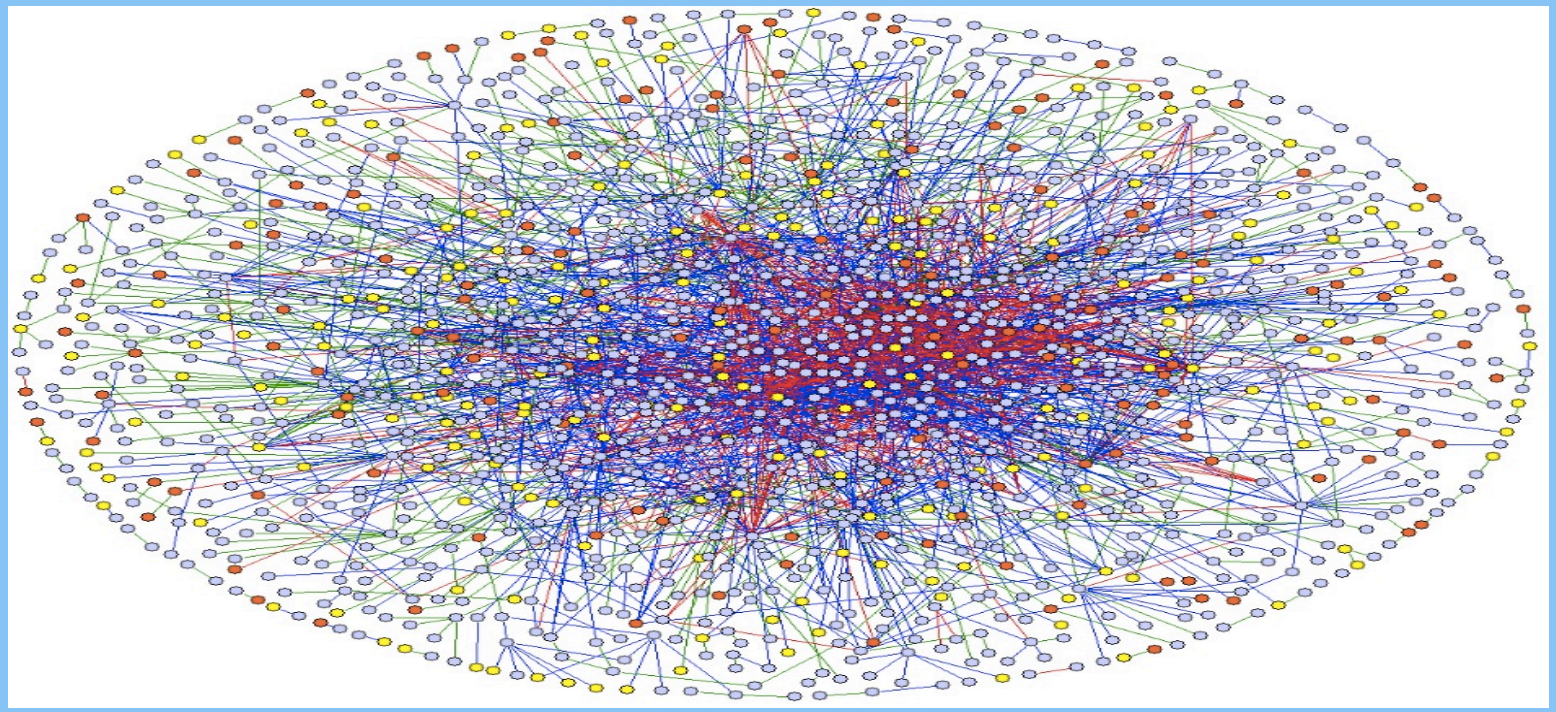


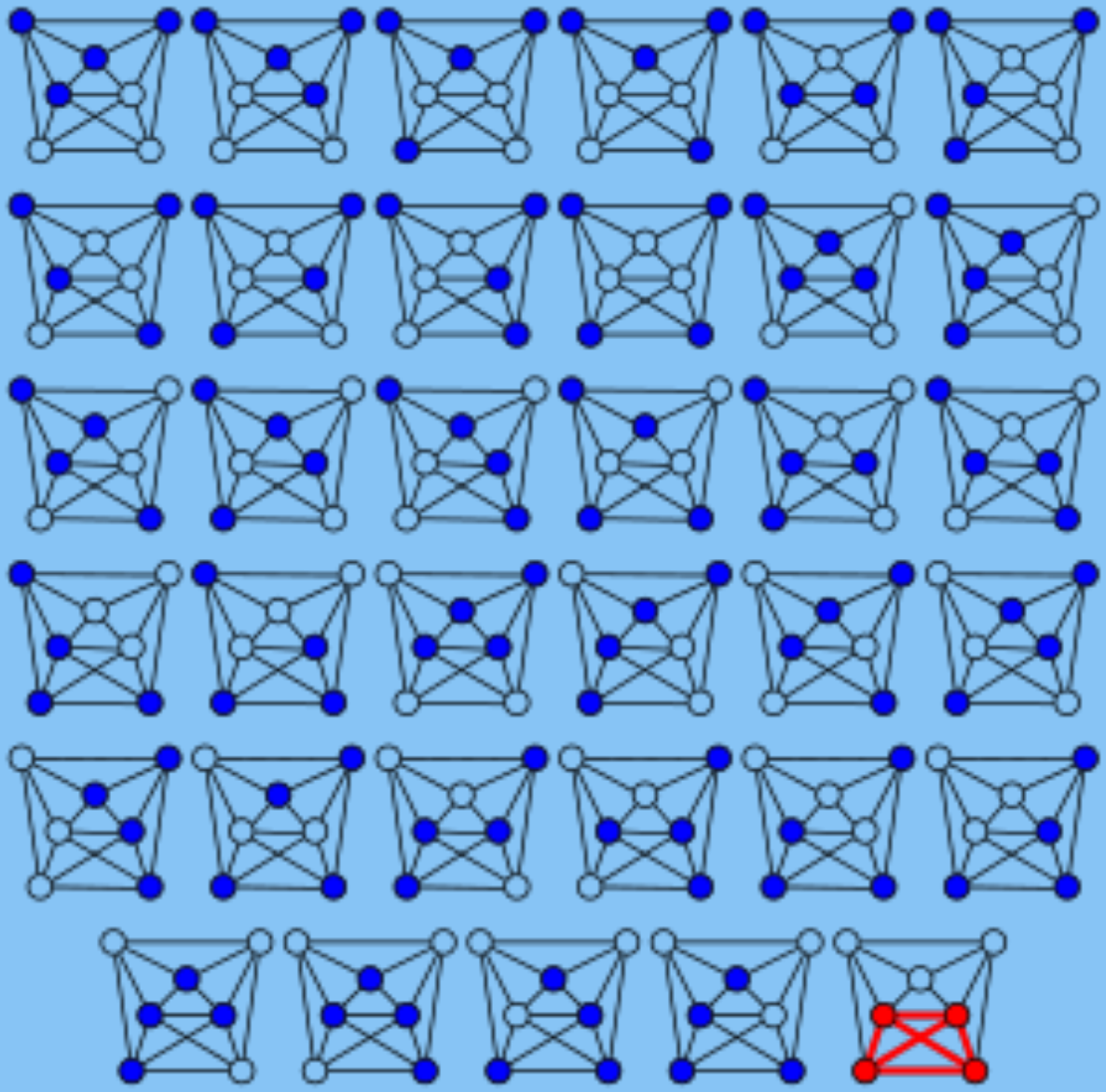




Un problema di CLIQUE più grande

Trovare la clique più grande in un grafo di 100 nodi può richiedere fino a secoli di tempo di calcolo con una ricerca di tutte le possibilità.





La ricerca esaustiva è necessaria ?

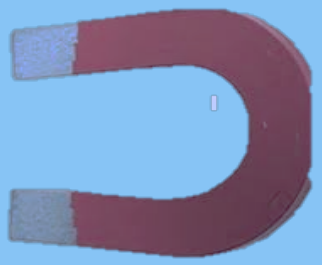
Non lo sappiamo.

Cercare un ago in un pagliaio

Trovato! Ci ho messo solo 10 giorni!

Cercare un ago…..

La ricerca esaustiva è necessaria?



No, se si ha a disposizione un magnete

Altri problemi difficili

- Scheduling
- Colorazione delle mappe
- Protein folding
- Isomorfismo di grafi
- Puzzle (Sudoku)
- Commesso viaggiatore
- Molti altri….

La questione di P e NP

Possiamo risolvere i problemi precedenti e altri che si risolvono facendo la ricerca esaustiva senza tale ricerca?

Non lo sappiamo

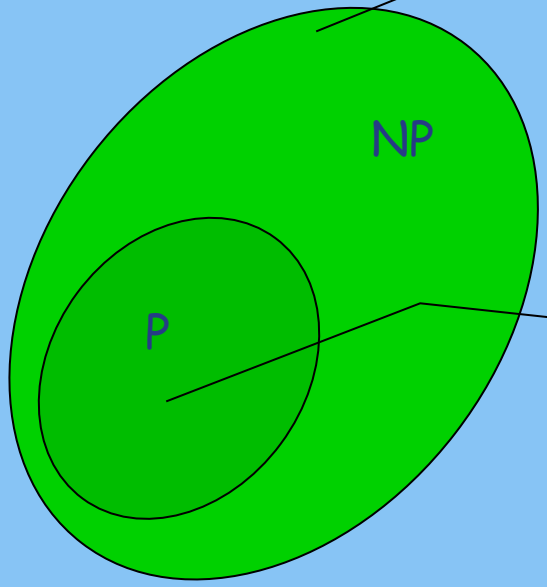
P e NP

- P "tempo Polinomiale"
  Problemi risolubili velocemente

- NP "tempo non deterministico Polinomiale"
  Problemi verificabili velocemente

  include i problemi precedenti

Le classi P e NP



Problemi facilmente verificabili

- Fattorizzazione
- Clique

Problemi facilmente risolvibili

- Moltiplicazione
- Ordinamento

P = NP

oppure

P ≠ NP

Storia della questione di P e NP

- 1960 Albori della teoria della complessità
  - Rabin, Blum, Hartmanis, Edmonds

- 1971 La questione di P e NP; NP-completezza
  - Cook, Levin, Karp

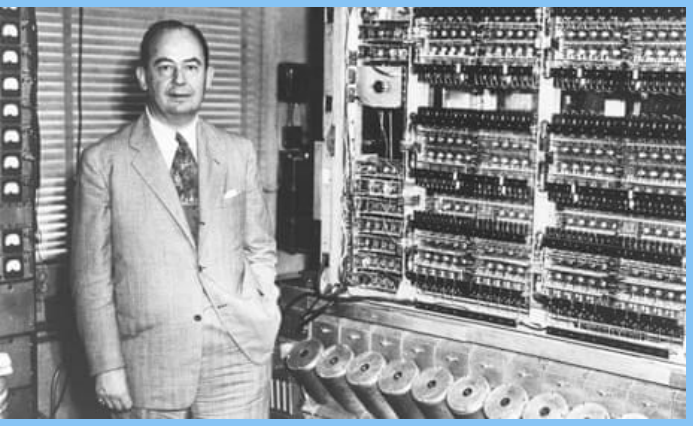
1956 Lettera di Gödel a Von Neuman
(scoperta negli anni 90)

- Anticipazione della questione P e NP

Kurt Gödel (1906 -1978)

- Il più grande logico del ventesimo secolo
- Risultati rivoluzionari nel 1930
- Anticipazione della questione P e NP

John Von Neumann (1903 -1957)



- Fondatore teoria dei giochi
- Pioniere del computer
- Moltissimi lavori fondazionali in matematica applicata, fisica e economia

Ma.. aveva il pallino delle bombe atomiche con cui voleva risolvere i problemi del mondo…

Ha ispirato il film di Kubrik il dottor Stranamore

Princeton, 20.III.1956

Lieber Herr v. Neumann!

Ich habe mit größtem Bedauern von Ihrer Erkrankung gehört. Die Nachricht kam nun ganz unerwartet. Morgenstern hatte zwar schon im Sommer von einem Schwächeanfall erzählt den Sie einmal hatten, aber er meinte damals, dass dem keine größere Bedeutung beizumessen sei. Wie ich höre, haben Sie sich in den letzten Monaten einer radikalen Behandlung unterzogen u. ich freue mich, dass diese den gewünschten Erfolg hatte u. es Ihnen jetzt besser geht. Ich hoffe u. wünsche Ihnen, dass Ihr Zustand sich bald noch weiter bessert u. dass die modernen Errungenschaften der Medizin, wenn möglich, zu einer vollständigen Heilung führen mögen.

Da Sie sich, wie ich höre, jetzt kräftiger fühlen, möchte ich mir erlauben, Ihnen über ein mathematisches Problem zu schreiben, über das mich

Ihr Ansicht kürzlich unterhielt: Man kann offenbar leicht eine Turingmaschine konstruieren, welche von jeder Formel F der engeren Funktionenkalküls u. jeder natürl. Zahl n zu entscheiden gestattet ob F einen Beweis der Länge n hat [Länge = Anzahl der Symbole]. Sei $\psi(F,n)$ die Anzahl der Schritte die die Maschine dazu benötigt u. sei $\varphi(n) = \max_F \psi(F,n)$. Die Frage ist, wie rasch $\varphi(n)$ für eine optimale Maschine wächst. Man kann zeigen $\varphi(n) \geq Kn$. Wenn es wirklich eine Maschine mit ~~$\varphi(n) \sim Kn$~~ $\varphi(n) \sim Kn$ (oder auch nur $\sim Kn^2$) gäbe, hätte das Folgerungen von der größten Tragweite. Es würde nämlich offenbar bedeuten, dass man trotz der Unlösbarkeit des Entscheidungsproblems die Denkarbeit der Mathematiker bei ja-oder-nein Fragen vollständig durch Maschinen ersetzen könnte. ~~Man müsste~~ Man müsste ja bloss das n so gross wählen, dass, wenn die Maschine kein Resultat liefert es auch keinen Sinn hat über die Problem nachzudenken. Nun scheint es mir aber durchaus im Bereich der Möglichkeit zu liegen, dass $\varphi(n)$ ist langsam wächst. Dann ist $\varphi(n) \geq Kn$ die einzige Abschätzung, die man durch eine Verallgemeinerung des Beweises für die Unlösbarkeit des Entscheidungsproblems erhalten kann; 2. bedeutet ja $\varphi(n) \sim Kn$ (oder $\sim Kn^2$) bloss, dass die Anzahl der Schritte gegenüber dem blossen Probieren von N auf $\log N$ (oder $(\log N)^2$) vermindert werden kann. Solche starke Verminderungen kommen aber bei andern finiten Problemen häufig vor, z.B. bei der Berechnung eines quadratischen Restzeichens durch wiederholte Anwendung des Reziprozitätsgesetzes. Es wäre interessant zu wissen, wie es damit z.B. bei der Feststellung, ob eine Zahl Primzahl ist, steht u. wie stark im allgemeinen bei finiten kombinatorischen Problemen die Anzahl der Schritte gegenüber dem blossen Probieren vermindert werden kann.

~~Ich~~ Ich weiss nicht ob Sie gehört haben, dass "Post's problem" (ob es unter den Problemen $(\exists y)\varphi(y,x)$ mit rekursivem φ Grade der Unlösbarkeit gibt) von einem ganz jungen Mann namens Richard Friedberg in positivem Sinn gelöst wurde. Die Lösung ist sehr elegant. Leider will Friedberg nicht Mathematik, sondern Medizin studieren (scheinbar unter dem Einfluss seines Vaters).

Was halten Sie übrigens von den Bestrebungen, die Analysis auf die verzweigte Typentheorie zu begründen, die neuerdings wieder in Schwung gekommen sind? Es ist Ihnen wahrscheinlich bekannt, dass Paul Lorenzen dabei bis zur Theorie der Lebesgueschen Masse vorgedrungen ist. Aber ich glaube, dass in wichtigen Teilen der Analysis nicht eliminierbare impredikative Schlussweisen vorkommen.

Ich würde mich sehr freuen, von Ihnen persönlich etwas zu hören; u. bitte lassen Sie es mich wissen, wenn ich irgend etwas für Sie tun kann.

Mit besten Grüssen u. Wünschen, auch an Ihre Frau Gemahlin

Ihr sehr ergebener Kurt Gödel

S. Ich gratuliere Ihnen herzlich zu den [illegible]

abgesehen von der Aufstellung der Axiome

Gödel aveva studiato la risolvibilità teorica dei problemi (decidibilità) e poneva nella lettera questioni sulla loro risolubilità pratica.

Chiedeva precisamente a Von Neumann con quante operazioni si riesce a sapere se un numero intero è primo e se uno dei problemi che si chiamano oggi "NP completi", secondo la definizione di Stephen Cook del 1971, poteva essere risolto in tempo polinomiale (lineare o quadratico)

Significato della questione di P e NP

Dobbiamo per forza fare la ricerca esaustiva quando abbiamo un problema come i precedenti?

A volte la ricerca esaustiva può essere evitata

Test di primalità

Un modo strano per vedere se un numero è primo

Vecchio teorema.

Per un primo p e un intero a < p:

$$a^{p-1} = 1 \pmod{p}$$

Esempi:

p=7, a=2: $2^6 = 64 = 1 \pmod 7$

p=15, a=2: $2^{14} = 16.384 = 4 \neq 1 \pmod{15}$

quindi 15 non è primo!

NP-completezza

NP-completezza

Problemi NP-completi :
Se uno è facile allora sono tutti facili!
Se uno è difficile allora sono tutti difficili!

Clique: NP-completo

Se Clique è in P allora P = NP

Colorazione delle mappe: NP-completo

Fattorizzazione: non si sa.

Se un problema NP-completo è in P allora P = NP

NP-completezza

Se un problema è NP-completo la speranza di trovare un algoritmo efficiente è molto bassa.

Se lo trovassimo non solo avremmo risolto il problema ma anche tutti gli altri problemi NP-completi con le riduzioni.

Tantissimi problemi NP-completi si trovano in matematica, biologia, fisica, economia...

Protein Engineering vol. 7 no. 9 pp. 1059-1068, 1994

The protein threading problem with sequence amino acid interaction preferences is NP-complete

Richard H. Lathrop

Economic Theory vol. 23, no. 2 , pp. 445-454, 2004

Finding a Nash equilibrium in spatial games is NP-complete

R. Baron, J. Durieu, H. Haller and P. Solal

[math.GR] arXiv:0802.3839v1

Quadratic equations over free groups are NP-complete

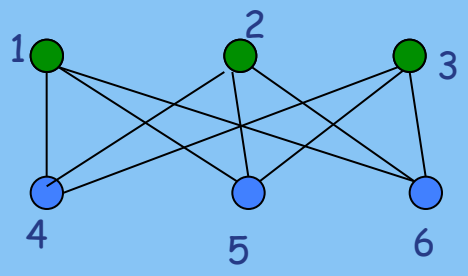
O. Kharlampovich, I.G. Lysenok, A G Myasnikov N. Touikan

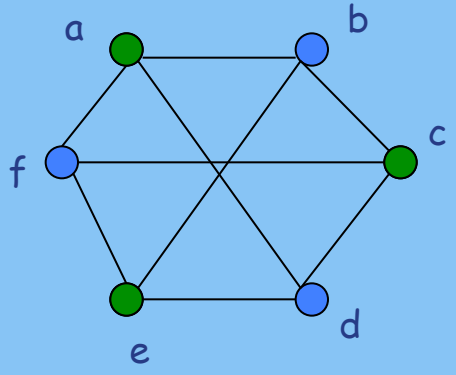
NP-completezza: segno di difficoltà.

Guida potenziale verso modelli e teorie migliori

Non tutti i problemi difficili sono nella classe NP.

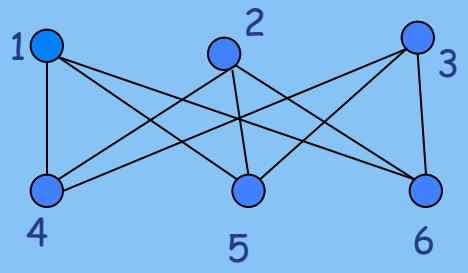
Isomorfismo o no?

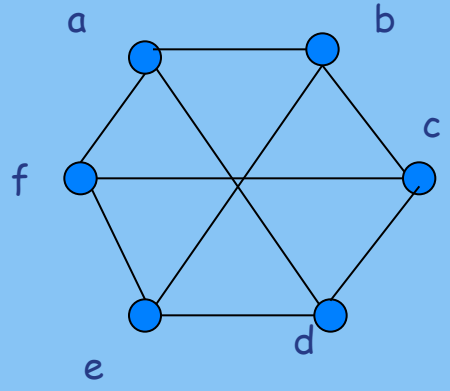




Sono isomorfi?

Facile da verificare



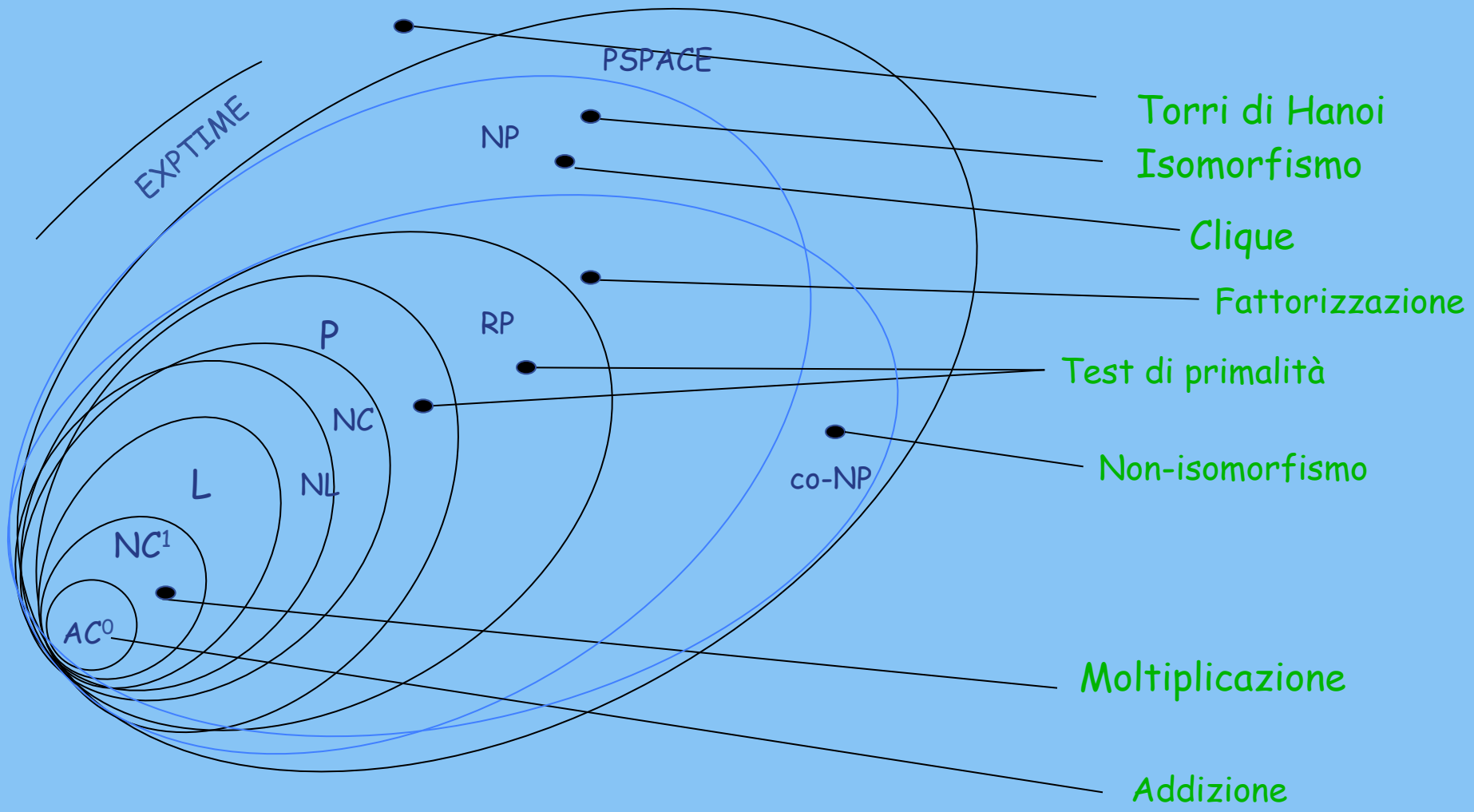


Sono non isomorfi?

Non è facile da verificare. Non-isomorfismo non appartiene a NP

Classi di complessità

Problemi:



Come provare che P ≠ NP ?

- Problema:

  Algoritmi molto sofisticati.

Bisognerebbe dimostrare che tutte le possibili strategie di risoluzione falliscono.

- Possibile strategie:

  Limitare le capacità della macchina

  Individuare gli input difficili

  input di grandi dimensioni (tendenti all'infinito)

Cosa succede in natura?

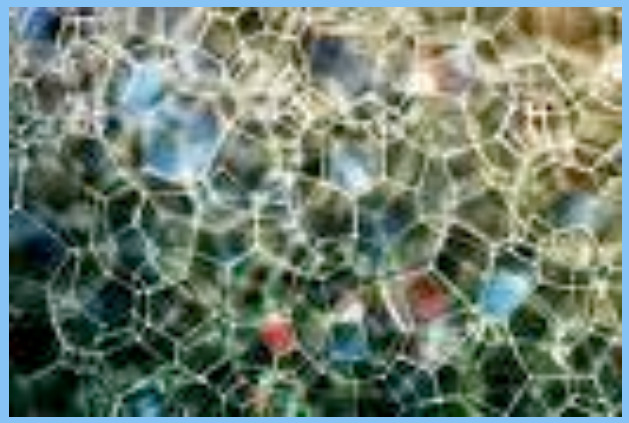
Problemi NP-completi che la "natura risolve"

Biologia: Protein Folding

Minima energia

Fisica: Schiuma

Minima superficie



Economia: Equilibrio di Nash in giochi strategici

Possibilità:

Il modello è sbagliato o sono input speciali o P=NP

Novità: Scienze Naturali ⟷ Informatica

Conseguenze positive di P≠NP

P≠NP Alcuni dei problemi che vogliamo risolvere sono difficili.

I problemi difficili sono utili?

Crittografia Se fattorizzazione è difficile:

- Codifica
- Firma digitale
- E-mail sicura
- Commercio elettronico
- Shopping on-line
- Poker on-line

Verrà mai risolto?

Servono nuove idee.

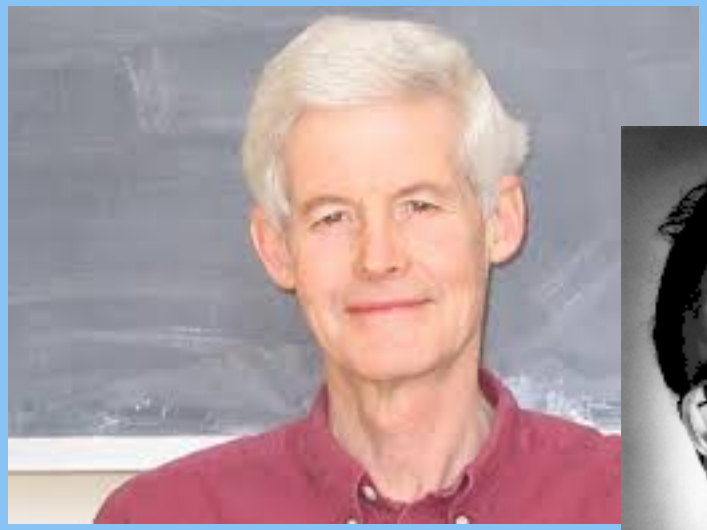
Cosa fare in pratica?

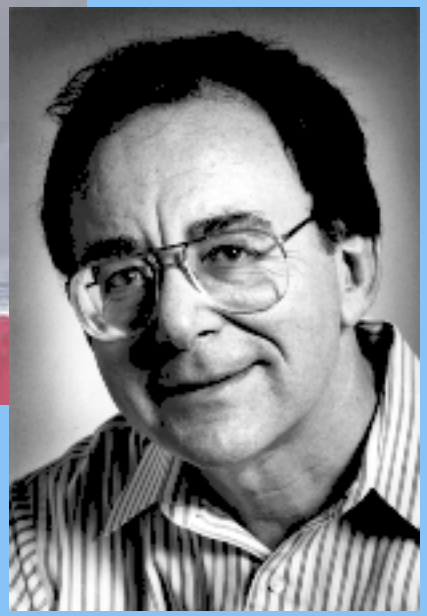
Il vostro datore di lavoro vi chiede di scrivere un programma veloce per un problema difficile.

Come rispondere?

Se il problema è NP-completo:

Io non lo so risolvere, ma non ci sono riusciti nemmeno tutti questi scienziati famosi…





Il problema è NP-completo se lo sapessi risolvere non sarei qui a lavorare!

Ma allora cosa fare?

Se la soluzione ottima è troppo difficile da ottenere, una soluzione quasi ottima ottenibile facilmente forse è buona abbastanza !

- Algoritmi approssimati.
  La soluzione ottenuta non è troppo lontana da quella ottima

- Algoritmi probabilistici.
  La soluzione ottima è ottenuta con alta probabilità

- Euristiche

  Simulazioni statistiche mostrano la bontà del metodo in molti casi

Il problema del resto automatico

Il problema del resto automatico

Una macchina automatica del caffè deve dare il resto all'utente, usando il minor numero di monete.

Proviamo una tecnica di algoritmo detta *greedy* (oppure *goloso* oppure *ingordo*).

Facciamo l'ingordi!

- Ordina le monete in ordine decrescente del valore e considerale nell'ordine.

- Se il resto da fornire è ≥ della moneta considerata, seleziona una moneta e aggiorna il resto ;

  altrimenti passa alla moneta successiva.

Algoritmo greedy

Esempio:

Resto = 165 centesimi: l'algoritmo seleziona 4 monete:

Prova con 2 euro : non è selezionata.

Prova con 1 euro = 100 centesimi: selezionata.
Nuovo resto = 65 centesimi;

Prova con 50 centesimi: selezionata.

Nuovo resto =15 centesimi;

Prova con 10 centesimi: selezionata.

Resto = 5 centesimi

Prova con 5 centesimi: selezionata.

Algoritmo greedy
È la soluzione ottima?

Tipo	**Quantità**
200	Illimitate
100	1
50	1
20	2
10	1
5	1
2	2
1	1

Numero massimo di monete per ogni tipo che possono essere incluse in una soluzione ottima

Algoritmo greedy

Soluzione ottima: minimo numero di monete

n_i = numero di monete da i centesimi nella sol. ottima

n_{200} = numero di monete da 200 centesimi

n_{100} = numero di monete da 100 centesimi

n_{50} = numero di monete da 50 centesimi

n_{20} = numero di monete da 20 centesimi

n_{10} = numero di monete da 10 centesimi

n_5 = numero di monete da 5 centesimi

n_2 = numero di monete da 2 centesimi

n_1 = numero di monete da 1 centesimo

Algoritmo greedy

Soluzione ottima: minimo numero di monete

Nella nostra soluzione $n_{200}=0$;

Dalla tabella sappiamo che

$n_{100}, n_{50}, n_{10}, n_5, n_1 \leq 1$ e

$n_{20}, n_2 \leq 2$;

Inoltre la sol.ottima non può usare 2 monete da 20 se ne ha usata una da 10, (una da 50 è meglio)

$10n_{10} + 20n_{20} \leq 40$ e

$n_1 + 2n_2 \leq 4$ (stesso motivo)

Algoritmo greedy

Soluzione ottima: minimo numero di monete

Il max valore restituito da una sol. ottima con monete inferiori a 100 centesimi

$n_1+2n_2+5n_5+10n_{10}+2n_{20}+50n_{50} \leq 4+5+40+50=99$

Dunque per dare come resto 165 centesimi, occorre una moneta 100 !

Restano 65 centesimi

Il max valore restituito da una sol. ottima con monete inferiori a 50 centesimi

$n_1+2n_2+5n_5+10n_{10}+2n_{20} < 4+5+40=49$

Occorre una moneta da 50 .....

Facciamo gli ingordi!

Algoritmo greedy: sceglie sempre la soluzione più vantaggiosa al momento.

Non sempre funziona!

In questo caso porta alla soluzione ottima.

Altri sistemi monetari non la garantiscono.

Facciamo gli ingordi!

monete da
100, 50, 30, 25, 10, 1 e 2

L'algoritmo greedy :
Resto 115
Seleziona una moneta 100, una da 10, 2 da 2 e una da 1. Numero di monete = 5

Soluzione ottima: 3 da 30, 1 da 25 = 4 monete.

Problema difficile! Sottile linea rossa…


GRAZIE