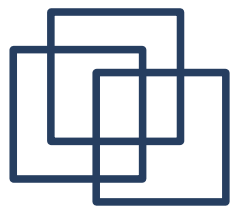


NP-Completezza

di

Andrea S. Gozzi
Valerio Romeo

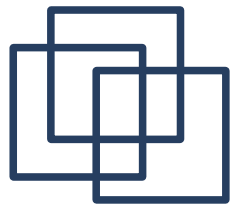


Argomenti trattati

Out of intense complexities, intense simplicities emerge.

Winston Churchill

- Concetti base & formalismi
- Introduzione alla Teoria della Complessità
- Classi di problemi
- Focus su NP e NP-Completezza
- Ipotesi su P vs NP
- Esempi



Un po' di terminologia

Cosa è un problema?

Relazione $P \subseteq I \times S$ dove I è l'insieme delle istanze ed S quello delle soluzioni.

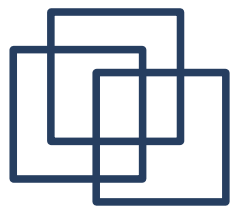
Cosa è un'istanza?

Un'istanza I di un problema P è un caso specifico del problema.

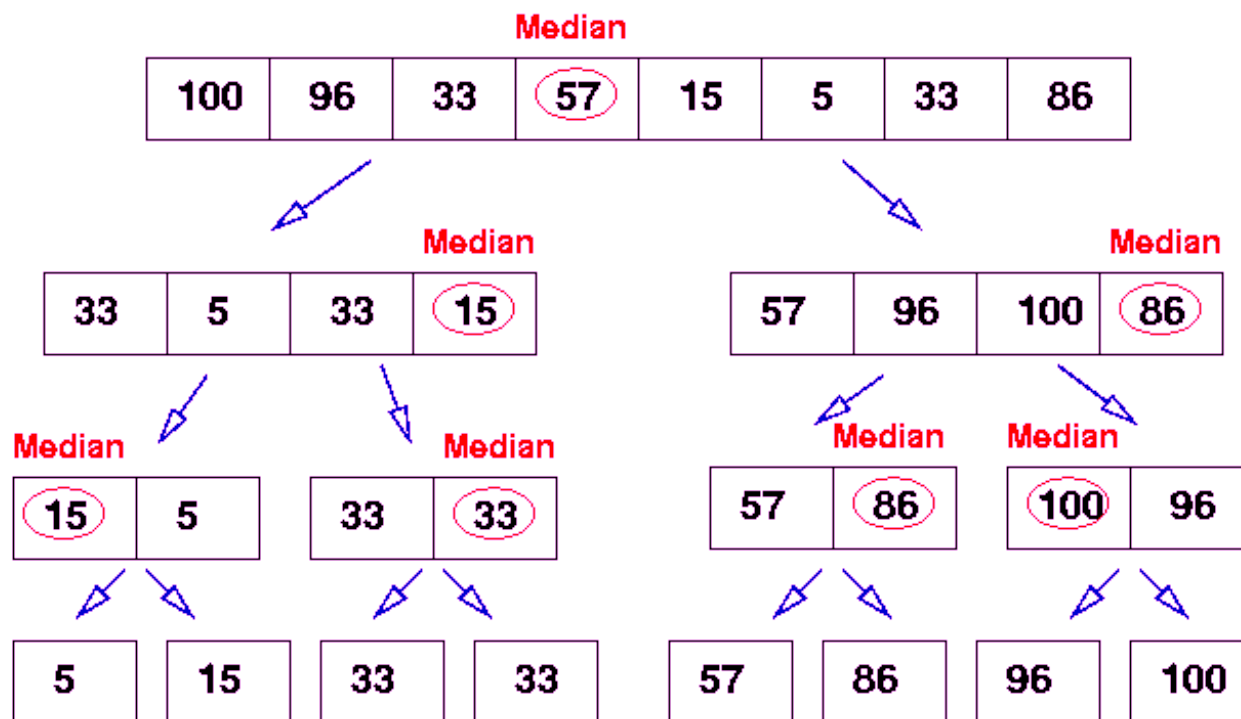
MT deterministiche/non-deterministiche

Nella teoria della computazione una macchina che, dato lo stesso ingresso e lo stesso stato interno, ammetta differenti uscite, è detta non-deterministica.

Una macchina che sempre, nel prendere una decisione, osservi un principio, sarebbe deterministica.

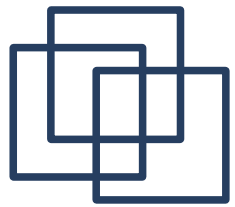


Quicksort (n^2)



Mediamente: **$O(n \cdot \log n)$**

Worst Case: **$O(n^2)$**

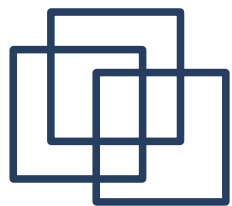


Teoria della complessità

– Classificazione di problemi in base alle risorse computazionali richieste per la loro soluzione.

TEMPO - M opera in tempo $f(n)$ se dato un input x di lunghezza n , la macchina M produce il risultato in $f(n)$ passi.

SPAZIO - M opera in spazio $f(n)$ se dato un input x di lunghezza n , la macchina M utilizza $f(n)$ celle "temporanee" per effettuare la computazione.



Tipi di problemi

- Problemi di decisione

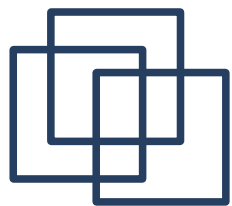
Soluzione: TRUE/FALSE



- Problemi di ottimizzazione

Soluzione: la migliore tra le possibili

- Problemi di enumerazione
- Problemi di ricerca



Classi di problemi

Classi principali

- P
- NP
- PSPACE
- NSPACE
- EXPTIME

$$P = \cup_{k>0} \text{TIME}(n^k)$$

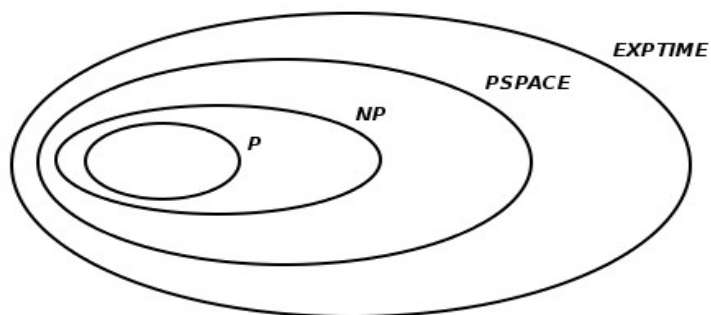
$$NP = \cup_{k>0} \text{NTIME}(n^k)$$

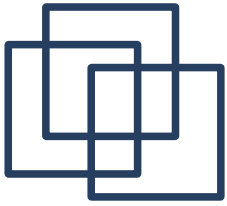
$$\text{PSPACE} = \cup_{k>0} \text{SPACE}(n^k)$$

$$\text{NSPACE} = \cup_{k>0} \text{NSPACE}(n^k)$$

$$\text{EXPTIME} = \cup_{k>0} \text{TIME}(2^{n^k})$$

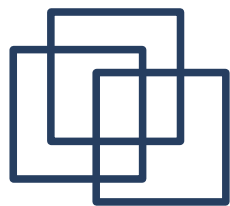
Possibili relazioni tra le classi:





Algoritmi

- algoritmi deterministici
 - algoritmi non deterministici
- Un algoritmo si dirà deterministico se per ogni istruzione esiste, a parità di dati d'ingresso, un solo passo successivo.
- Invece, non deterministico se contiene almeno una istruzione che ammette più passi successivi.



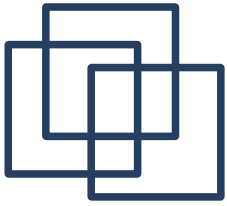
NP (1)

Vogliamo verificare se un'istanza w soddisfa una certa proprietà in un problema decisionale P .

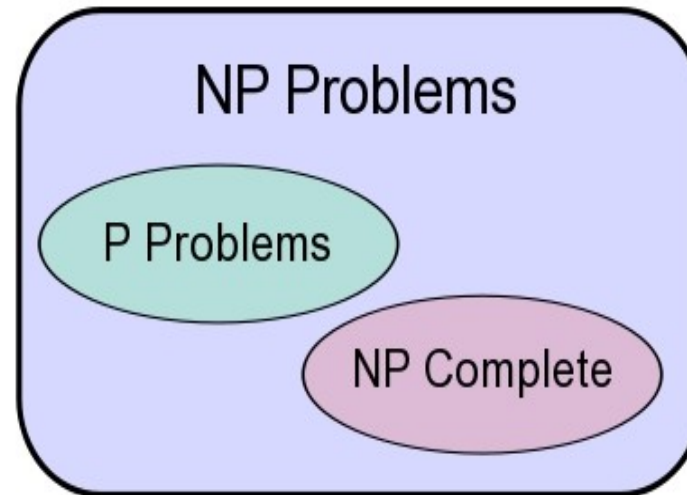
Bisogna quindi trovare un oggetto $K(w, P)$ che dimostri la proprietà soddisfatta.

 **Certificato**

I problemi contenuti nella classe NP ammettono certificati verificabili in tempo polinomiale.

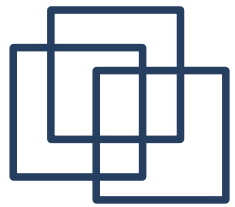


NP (2)



Un problema **P1** é definito “NP-Completo” quando:

- **$P1 \in NP$**
- per ogni **$P2 \in NP$** allora **$P2 \leq_p P1$**



Riducibilità polinomiale

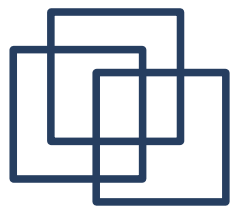
Sia $\mathbf{P1}$ e $\mathbf{P2} \in \mathbf{NP}$, $\mathbf{P1}$ si riduce in tempo polinomiale a $\mathbf{P2}$ ($\mathbf{P1} \leq \mathbf{P2}$) se esiste un algoritmo per risolvere $\mathbf{P1}$ che chiama un certo numero di volte un ipotetico algoritmo per $\mathbf{P2}$, e risulta polinomiale se si suppone che quello per $\mathbf{P2}$ richieda un'unica unità di tempo.



NP-Completezza

Come detto in precedenza, i problemi NP-completi non ammettano algoritmi polinomiali di risoluzione.

Basterebbe infatti provare che uno qualsiasi di essi é risolvibile in tempo polinomiale, perché tutti i problemi in NP siano risolvibili in tempo polinomiale.



P=NP?

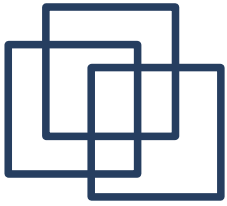


CLAY MATHEMATICS INSTITUTE

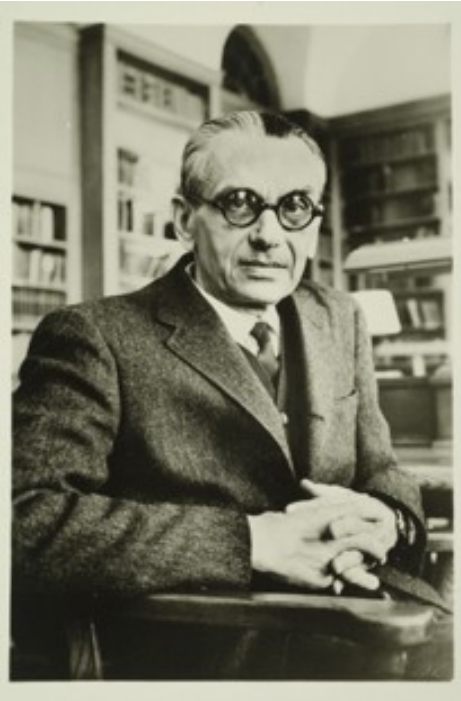
Dedicated to increasing and disseminating mathematical knowledge

Nel settembre 2000 il Clay Institute (USA) ha istituito un fondo per premiare chiunque riuscisse a risolvere vari problemi matematici, tra cui il “**P=NP?**”.

Fino ad oggi nessuno ha ancora riscosso il milione di dollari in palio...

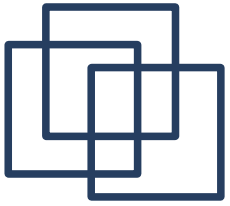


P = NP (1)



K. Gödel (1906-1978)

- É possibile costruire una MT che decida se per un numero naturale \mathbf{n} ed una teoria del primo ordine \mathbf{F} esiste una “**proof of length**” di \mathbf{n} .
- $\psi(\mathbf{F}, \mathbf{n})$ numero di passi $\longrightarrow \gamma(\mathbf{n}) = \max(\mathbf{F}, \mathbf{n})$
- Si può dimostrare come $\gamma(\mathbf{n}) \geq \mathbf{K} \cdot \mathbf{n}$, ma quanto cresce $\gamma(\mathbf{n})$?



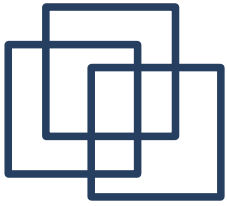
P = NP (2)

Una soluzione lineare o polinomiale per un input n , sembra essere “quite within the realm of possibilities”.

Questo però significherebbe che:

“the reasoning of mathematicians about yes-or-no questions can be completely replaced by machines.”.

Un modesto informatico, dimostrando che $P=NP$, potrebbe quindi vincere tutti e 7 i premi offerti dal Clay Institute!



Se $P=NP$...

- Metodi crittografici ora sicuri diventano facilmente violabili.
- Precisione vicina al 100% nel riconoscimento visivo automatizzato, nelle traduzioni e in genere per tutti i problemi di apprendimento automatico.
- Miglioramenti significativi nella previsione del meteo e dei fenomeni naturali.
- Trasporti per persone e materiali più veloci ed economici.

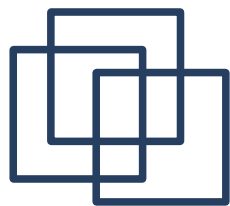


$P \neq NP$ (1)

I teorici moderni pensano invece che **$P \neq NP$** , principalmente perché nessuna prova del contrario é stata trovata in decenni di ricerche.

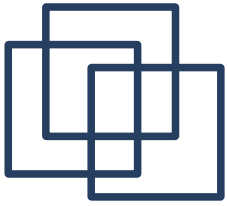
“The main argument in favour of $P \neq NP$ is the total lack of fundamental progress in the area of exhaustive search. ”

Moshe Vardi, Rice University



$P \neq NP$ (2)

Nonostante le implicazioni di una possibile dimostrazione di $P \neq NP$ non avrebbero lo stesso impatto di $P = NP$, si tratterebbe comunque di un sostanziale avanzamento nella teoria computazionale e permetterebbe di focalizzare gli sforzi per la risoluzione parziale dei problemi NP o di altri problemi notevoli.



SAT

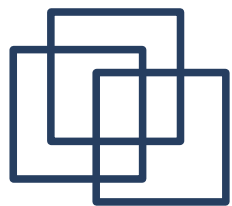
- Istanza: Formula Booleana
- Problema: Verificare la soddisfacibilità della formula

Formula soddisfattibile:

$$((F \vee T \vee \neg T) \wedge \neg F) \vee \neg(T \wedge T)$$

Formula non soddisfattibile:

$$X_1 \wedge \neg X_1$$

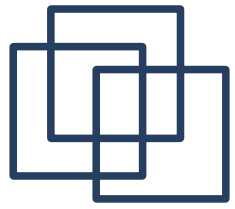


Minesweeper è NP (1)



“Minesweeper” é un videogioco per PC per singolo giocatore inventato da Robert Donner nel 1989. Lo scopo del gioco é ripulire un campo minato senza far esplodere le mine.

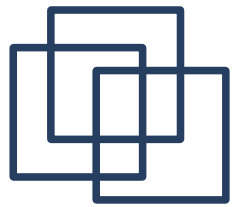
Se giocata con abilità, una partita può essere vinta senza rischiare troppi tentativi casuali.



Minesweeper é NP (2)

Giocare “abilmente” significa individuare prima di tutto le caselle libere, e quindi concentrarsi su quelle potenzialmente minate.

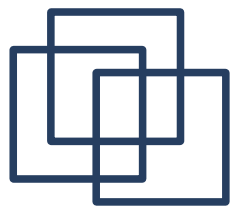
Per determinare se una casella sia innocua, la si indica come minata e si verifica se la configurazione sia consistente: se la risposta è negativa, allora la cella può essere scoperta tranquillamente!



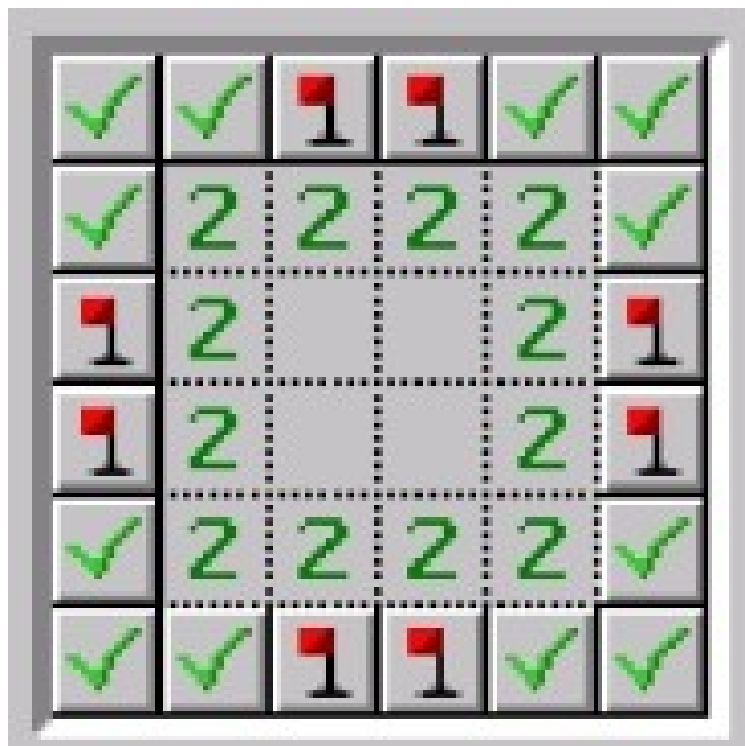
Minesweeper é NP (2)

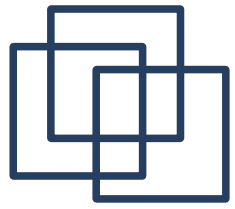


Determinare se nella casella (2,6) ci sia una mina non é particolarmente complicato ma con una configurazione diversa...



Minesweeper é NP (2)

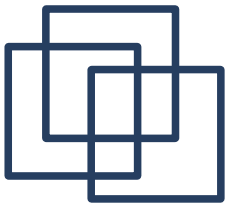




Minesweeper é NP (3)

Si capisce chiaramente che Minesweeper appartiene alla classe NP, infatti per valutare la consistenza di una configurazione é necessario verificare ogni possibilità.

Riducendo un problema SAT in una board Minesweeper permette di dimostrare come sia anche NP-Completo.

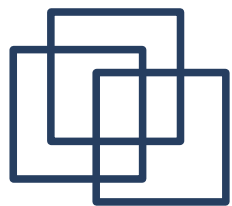


Sudoku (1)

Il Sudoku é un gioco di logica nel quale al giocatore o solutore viene proposta una griglia di 9×9 celle.

3	5	4	1	8	6	2	7	9
1	9	6	2	4	7	3	5	8
8	2	7	3	5	9	1	6	4
9	7	5	8	2	1	4	3	6
6	1	8	7	3	4	9	2	5
4	3	2	6	9	5	8	1	7
5	8	1	9	6	2	7	4	3
7	4	9	5	1	3	6	8	2
2	6	3	4	7	8	5	9	1

L'obiettivo del gioco consiste nel riempire il diagramma in modo che in tutte le righe orizzontali, in tutte le colonne verticali e in tutti quadrati 3×3 compaiano una sola volta i numeri da 1 a 9.



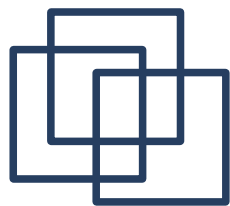
Sudoku (2)

Generalizzando il problema, una istanza del Sudoku é una tavola \mathbf{G} $n^2 \cdot n^2$, suddivisa in n^2 quadratini $n \cdot n$, contenente alcuni interi compresi tra 1 e n^2 .

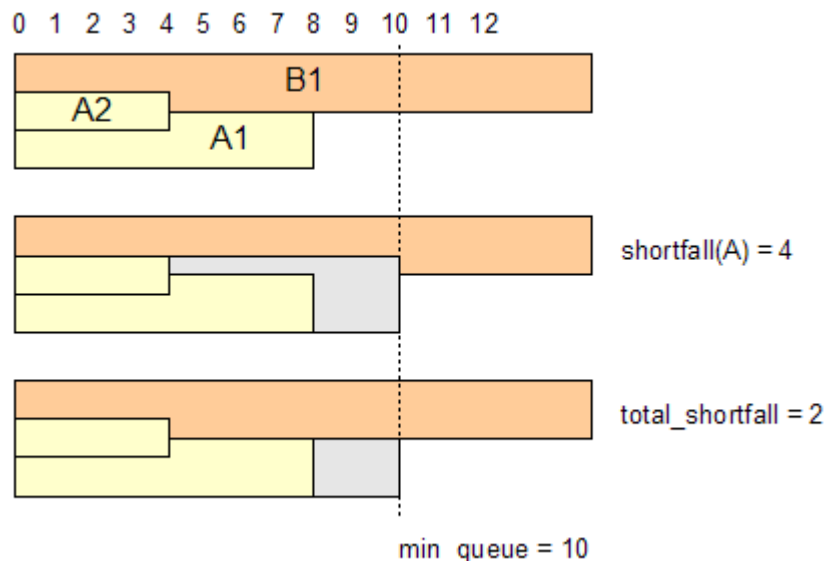
Il peso della istanza é n .

La domanda é: "si può completare \mathbf{G} in modo tale che in ogni riga ed in ogni colonna di \mathbf{G} gli interi tra 1 e n^2 appaiano una ed una sola volta, e siano al tempo stesso tutti presenti in ogni quadratino?"

Ovviamente il problema sta in NP.

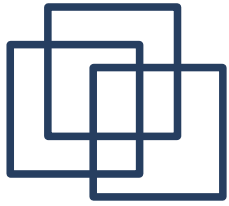


Job Scheduling



Dati J processi dove L_i é il tempo necessario al completamento di J_i .
Si vuole suddividerli su M macchine in modo che vengano completati nel minor tempo possibile riducendo il *makespan*.

Infatti non può essere trovato uno scheduler ottimo per multiprocessori senza una conoscenza a priori delle deadline, tempi di computazione e tempi di arrivo di tutti i task.



Bibliografia

- **Minesweeper and NP-completeness** – Richard W. Kaye
- **April 1989 Structural Complexity Column** – J. Hartmanis
- **Le Scienze Agosto 2005** – P. Odifreddi
- **Un premio per Gauss** – U. Cerruti
- **Complexity and completeness of finding another solution and its application to puzzle** – T. Yato
- **Complexity 2003** - D.Moshk