Qualche riflessione a margine...

Claudio Mirolo

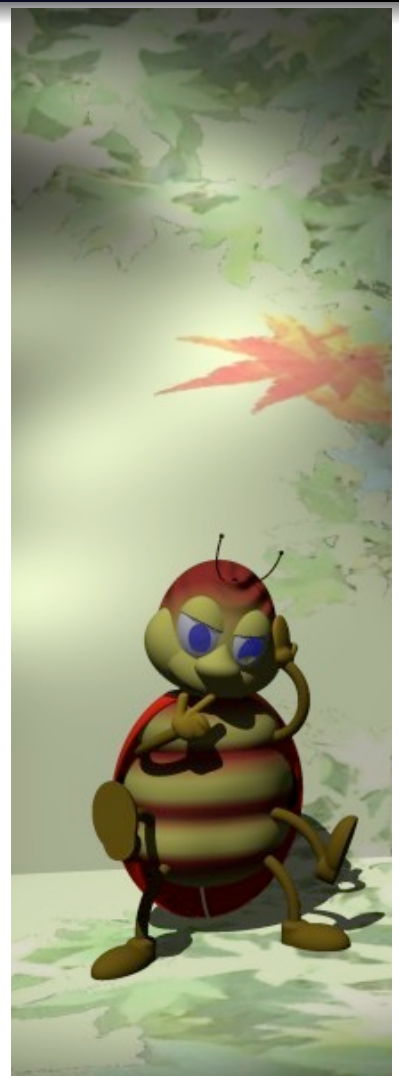
Dipartimento di Scienze Matematiche, Informatiche e Fisiche,
Università di Udine, via delle Scienze 206 – Udine

claudio.mirolo@uniud.it

Udine, 14 dicembre 2018

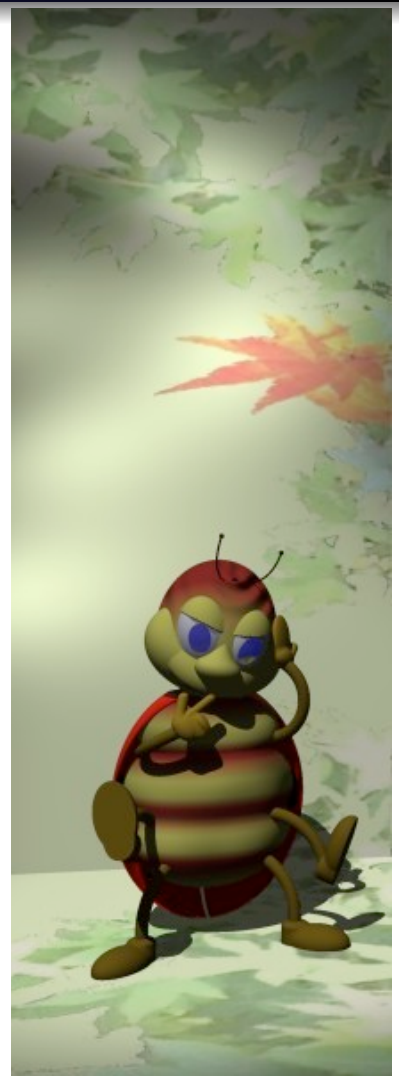
P = NP ... ?

- Di cosa stiamo parlando?
- Problema importante dal punto di vista teorico
  ... ma sarò impreciso al riguardo
- Ci sono però risvolti estremamente concreti
  ... proviamo a coglierne il senso



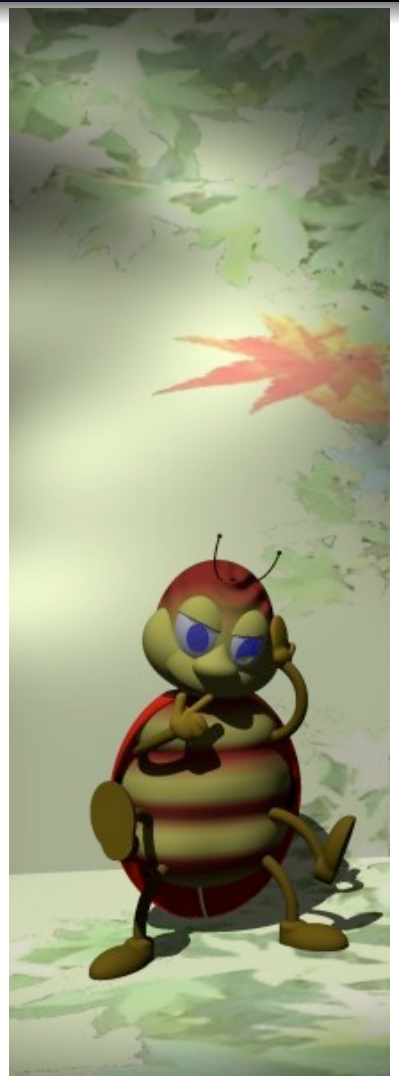
P = NP ... ?

- Di cosa stiamo parlando?
- Problema importante dal punto di vista teorico
  ... ma sarò impreciso al riguardo
- Ci sono però risvolti estremamente concreti
  ... proviamo a coglierne il senso



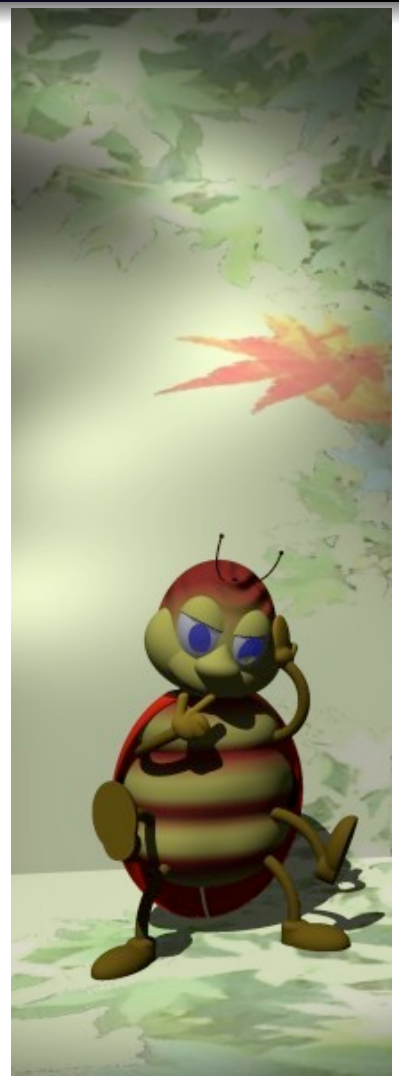
P = NP ... ?

- Di cosa stiamo parlando?
- Problema importante dal punto di vista teorico
  ... ma sarò impreciso al riguardo
- Ci sono però risvolti estremamente concreti
  ... proviamo a coglierne il senso



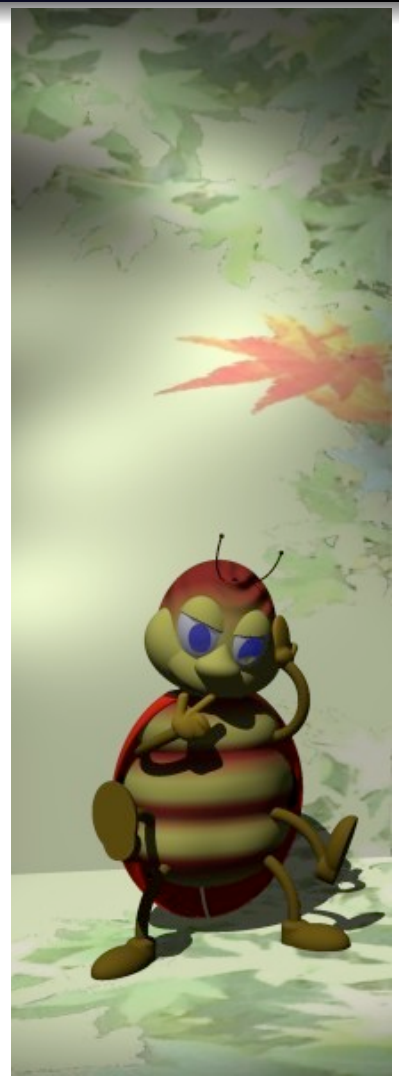
P = NP ... ?

- Di cosa stiamo parlando?
- Problema importante dal punto di vista teorico
  ... ma sarò impreciso al riguardo
- Ci sono però risvolti estremamente concreti
  ... proviamo a coglierne il senso



P = NP ... ?

- Di cosa stiamo parlando?
- Problema importante dal punto di vista teorico
  ... ma sarò impreciso al riguardo
- Ci sono però risvolti estremamente concreti
  ... proviamo a coglierne il senso



P = NP ... ?

- Si fa riferimento al “tempo di calcolo”
- Laciamo per un po’ da parte **P** ed **NP**

P = NP ... ?

- Si fa riferimento al “tempo di calcolo”
- Laciamo per un po’ da parte **P** ed **NP**

Sequenze di bit

- Proviamo a seguire questo percorso. . .
- Quante sequenze *diverse* di 2 bit?

Sequenze di bit

- Proviamo a seguire questo percorso. . .

- Quante sequenze *diverse* di 2 bit?

Sequenze di due bit

Sequenze di due bit

00

Sequenze di due bit

00

01

Sequenze di due bit

00

01

10

Sequenze di due bit

00

01

10

11

Sequenze di un bit

- E quante di 1 bit?

Sequenze di un bit

- E quante di 1 bit?

0

1

Sequenze di tre bit

- E ancora quante sequenze diverse di 3 bit?

Sequenze di tre bit

Abbiamo visto: 4 sequenze di due bit. . .

Sequenze di tre bit

Abbiamo visto: 4 sequenze di due bit...

00

01

10

11

Sequenze di tre bit

E poi, per tre bit. . .

000

001

010

011

Sequenze di tre bit

E poi, per tre bit. . .

000	100
001	100
010	100
011	100

Sequenze di *n* bit

Dunque...

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di *n* bit

Dunque...

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di *n* bit

Dunque...

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di *n* bit

Dunque...

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di *n* bit

Dunque. . .

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di *n* bit

Dunque. . .

- 2 sequenze diverse da 1 bit
- 4 sequenze diverse da 2 bit
- 8 sequenze diverse da 3 bit
- ? sequenze diverse da 4 bit
- ? sequenze diverse da 5 bit

Sequenze di 64 bit

- Se aggiungo un bit, il numero di sequenze diverse . . . ?
- Quante sequenze diverse di 64 bit?
- 64 bit non sono poi così tanti!

Sequenze di 64 bit

- Se aggiungo un bit, il numero di sequenze diverse . . . ?
- Quante sequenze diverse di 64 bit?
- 64 bit non sono poi così tanti!

Sequenze di 64 bit

- Se aggiungo un bit, il numero di sequenze diverse . . . ?
- Quante sequenze diverse di 64 bit?
- 64 bit non sono poi così tanti!

Sequenze di 64 bit

- Se aggiungo un bit, il numero di sequenze diverse . . . ?
- Quante sequenze diverse di 64 bit?
- 64 bit non sono poi così tanti! (aritmetica floating-point)

Sequenze di 64 bit

- $2^{64} > 1.8 \cdot 10^{19}$
- Immaginiamo che vogliamo vedere le sequenze una a una, ciascuna per un secondo...
- Quanti secondi?

Sequenze di 64 bit

- $2^{64} > 1.8 \cdot 10^{19}$
- Immaginiamo che vogliamo vedere le sequenze una a una, ciascuna per un secondo…
- Quanti secondi?

Sequenze di 64 bit

- $2^{64} > 1.8 \cdot 10^{19}$
- Immaginiamo che vogliamo vedere le sequenze una a una, ciascuna per un secondo…
- Quanti secondi?

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? . . . $2.1 \cdot 10^{14}$ giorni
- Quanti anni? . . . $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Sequenze di 64 bit

- $1.8 \cdot 10^{19}$ sec
- Quanti minuti? Quante ore?
- Quanti giorni? ... $2.1 \cdot 10^{14}$ giorni
- Quanti anni? ... $5.8 \cdot 10^{11}$ anni

Tempi di calcolo

- E adesso immaginiamo che l'unico "algoritmo" che conosciamo per determinare quale sequenza di bit risolve un certo problema verifichi le sequenze una per una
- Certo, il computer è molto più veloce...
  può testare una sequenza in un ns (miliardesimo di sec)!
- Quanti anni? ... $5.8 \cdot 10^2$ anni

Tempi di calcolo

- E adesso immaginiamo che l'unico "algoritmo" che conosciamo per determinare quale sequenza di bit risolve un certo problema verifichi le sequenze una per una
- Certo, il computer è molto più veloce...
  può testare una sequenza in un ns (miliardesimo di sec)!
- Quanti anni? ... $5.8 \cdot 10^2$ anni

Tempi di calcolo

- E adesso immaginiamo che l'unico "algoritmo" che conosciamo per determinare quale sequenza di bit risolve un certo problema verifichi le sequenze una per una
- Certo, il computer è molto più veloce...
  può testare una sequenza in un ns (miliardesimo di sec)!
- Quanti anni? ... $5.8 \cdot 10^2$ anni

Tempi di calcolo

- E adesso immaginiamo che l'unico "algoritmo" che conosciamo per determinare quale sequenza di bit risolve un certo problema verifichi le sequenze una per una
- Certo, il computer è molto più veloce...
  può testare una sequenza in un ns (miliardesimo di sec)!
- Quanti anni? ... $5.8 \cdot 10^2$ anni

Tempi di calcolo

- E adesso immaginiamo che l'unico "algoritmo" che conosciamo per determinare quale sequenza di bit risolve un certo problema verifichi le sequenze una per una
- Certo, il computer è molto più veloce...
  può testare una sequenza in un ns (miliardesimo di sec)!
- Quanti anni? ... $5.8 \cdot 10^2$ anni

Tempi di calcolo

- Ci sono problemi per cui la situazione si presenta più o meno in questi termini...
- Per trovare una soluzione non so fare altro che esplorare, caso per caso, un gigantesco spazio di ricerca...
  ... p. es. tutte le 2^{64} sequenze di 64 bit
- Per verificare se una data sequenza rappresenta una soluzione, è invece sufficiente scandirne i bit uno a uno
  ... magari in appena 64 nsec!

Tempi di calcolo

- Ci sono problemi per cui la situazione si presenta più o meno in questi termini...
- Per trovare una soluzione non so fare altro che esplorare, caso per caso, un gigantesco spazio di ricerca...
  ... p. es. tutte le 2^{64} sequenze di 64 bit
- Per verificare se una data sequenza rappresenta una soluzione, è invece sufficiente scandirne i bit uno a uno
  ... magari in appena 64 nsec!

Tempi di calcolo

- Ci sono problemi per cui la situazione si presenta più o meno in questi termini...
- Per trovare una soluzione non so fare altro che esplorare, caso per caso, un gigantesco spazio di ricerca...
  ... p. es. tutte le 2^{64} sequenze di 64 bit
- Per verificare se una data sequenza rappresenta una soluzione, è invece sufficiente scandirne i bit uno a uno
  ... magari in appena 64 nsec!

Tempi di calcolo

- Ci sono problemi per cui la situazione si presenta più o meno in questi termini...
- Per trovare una soluzione non so fare altro che esplorare, caso per caso, un gigantesco spazio di ricerca...
  ... p. es. tutte le 2^{64} sequenze di 64 bit
- Per verificare se una data sequenza rappresenta una soluzione, è invece sufficiente scandirne i bit uno a uno
  ... magari in appena 64 nsec!

Tempi di calcolo

- Ci sono problemi per cui la situazione si presenta più o meno in questi termini...
- Per trovare una soluzione non so fare altro che esplorare, caso per caso, un gigantesco spazio di ricerca...
  ... p. es. tutte le 2^{64} sequenze di 64 bit
- Per verificare se una data sequenza rappresenta una soluzione, è invece sufficiente scandirne i bit uno a uno
  ... magari in appena 64 nsec!

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  . . . e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra. . .
- (circa)

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  . . . e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra. . .
- (circa)

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  . . . e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra. . .
- (circa)

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  ... e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra...
- (circa)

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  ... e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra...
- (circa)

NP

- La classe **NP** comprende problemi di questo tipo
- Se avessi un numero illimitato di unità di calcolo (CPU) potrei distribuire le 2^{64} sequenze ad altrettante CPU
  ... e in 64 nsec una di esse troverà la soluzione
- **NP** = **P**olinomiale **N**on-deterministico
- Ma non si conosce un algoritmo per stare meno di $5.8 \cdot 10^2$ anni, eseguendo una operazione dopo l'altra...
- (circa)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k (k fissato una volta per tutte)
- . . . E se $k = 11$? (tutte le combinazioni di 11 posizioni dei bit)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k (k fissato una volta per tutte)
- . . . E se $k = 11$? (tutte le combinazioni di 11 posizioni dei bit)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k (k fissato una volta per tutte)
- . . . E se $k = 11$? (tutte le combinazioni di 11 posizioni dei bit)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k (k fissato una volta per tutte)
- . . . E se $k = 11$?
  (tutte le combinazioni di 11 posizioni dei bit)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k (k fissato una volta per tutte)
- . . . E se $k = 11$? (tutte le combinazioni di 11 posizioni dei bit)

P

- I problemi che si possono *concretamente* trattare rientrano nella classe **P** = **P**olinomiale (deterministico)
- Ma attenzione: polinomiale non vuol dire necessariamente lineare – tempo che cresce in proporzione al numero di bit
- Ma, più in generale, tempo che cresce come n^k
  (k fissato una volta per tutte)
- . . . E se $k = 11$?
  (tutte le combinazioni di 11 posizioni dei bit)

Epilogo

- La computazione del futuro sarà quantistica?
- Oppure ad elevatissimo livello di parallelismo? (via cloud)
- Oppure assumerà qualche altra forma?

Epilogo

- La computazione del futuro sarà quantistica?
- Oppure ad elevatissimo livello di parallelismo? (via cloud)
- Oppure assumerà qualche altra forma?

Epilogo

- La computazione del futuro sarà quantistica?
- Oppure ad elevatissimo livello di parallelismo? (via cloud)
- Oppure assumerà qualche altra forma?

Epilogo

- La computazione del futuro sarà quantistica?
- Oppure ad elevatissimo livello di parallelismo? (via cloud)
- Oppure assumerà qualche altra forma?