Algoritmi e Strutture Dati

Luciano Gualà
guala@mat.uniroma2.it
www.mat.uniroma2.it/~guala

Picture-Hanging Puzzles

Algoritmi ricorsivi e equazioni di ricorrenza: uno scenario meno comune

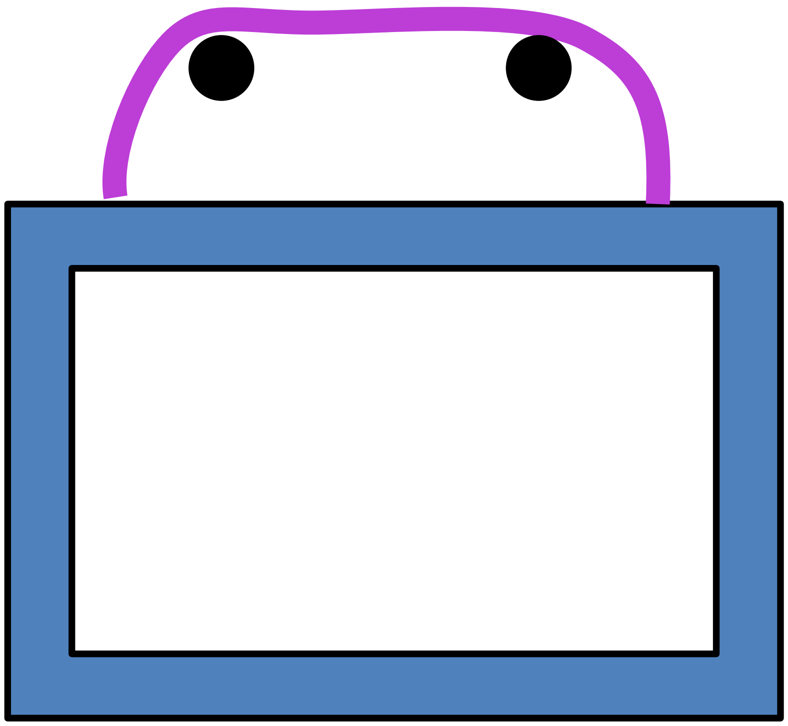
[riferimento:]

E. Demaine, M. Demaine, Y. Minsky, J.Mitchell, R. Rivest, M. Patrascu,
Picture-Hanging Puzzles, FUN'12

Un modo perverso di attaccare quadri: puzzle, matematica, algoritmi

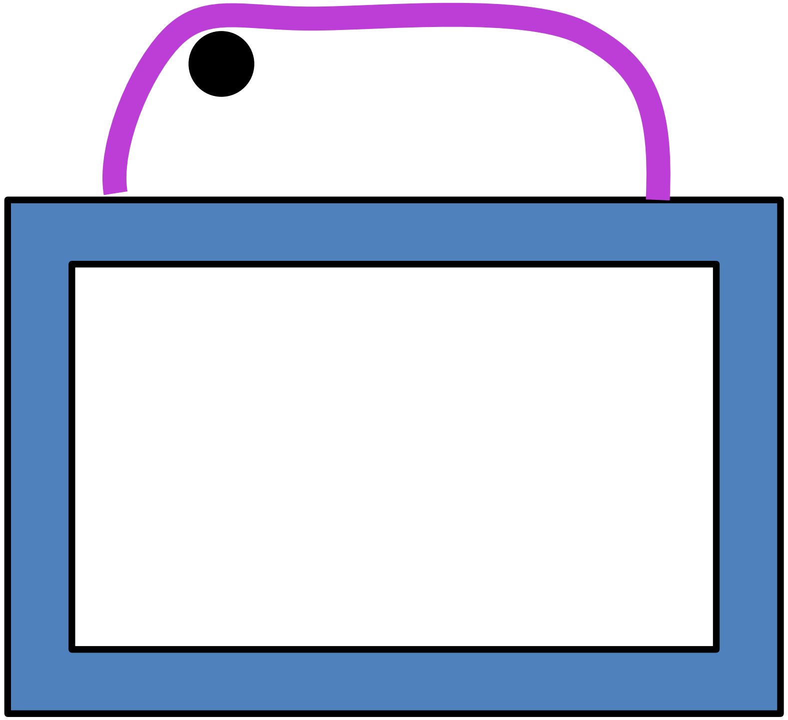
…e un paio di cose che ho imparato sull'informatica

Un modo classico di appendere un quadro:



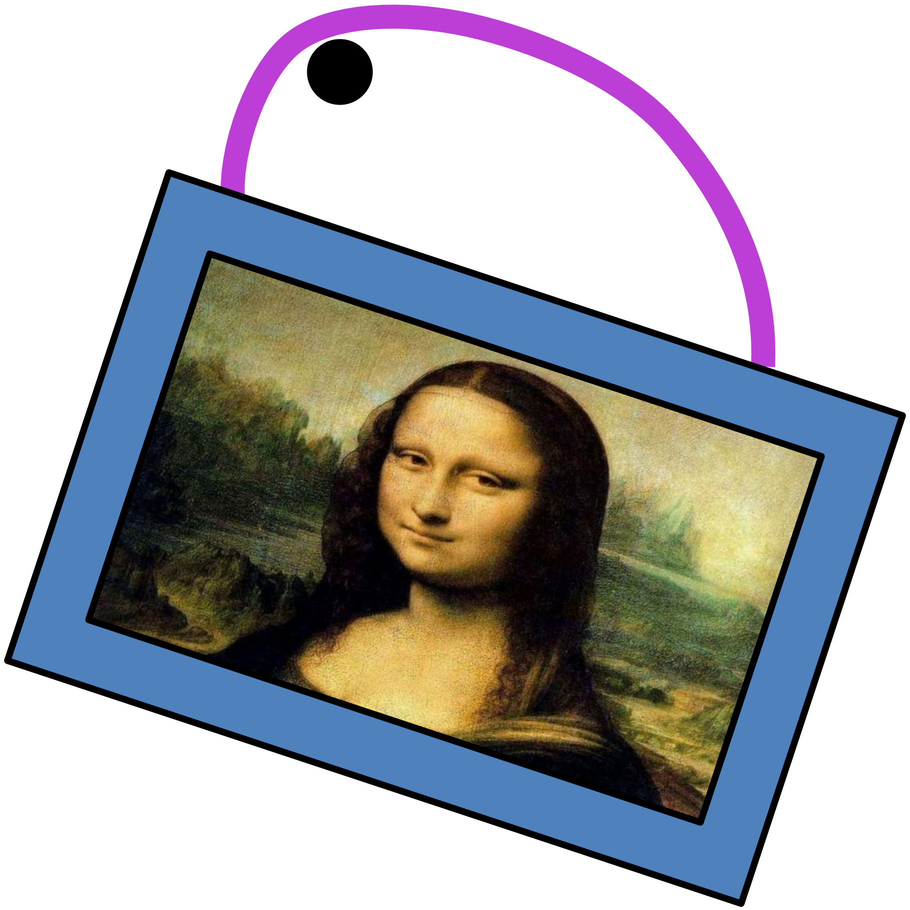
Che succede al quadro se rimuoviamo un chiodo?

Un modo classico di appendere un quadro:



Che succede al quadro se rimuoviamo un chiodo?

Un modo classico di appendere un quadro:



Che succede al quadro se rimuoviamo un chiodo?

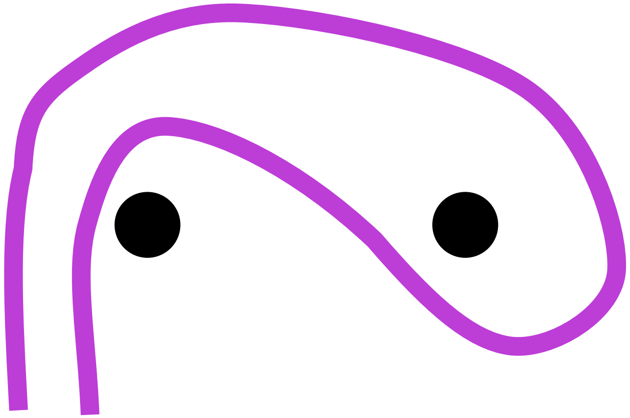
niente: il quadro resta appeso sull'altro chiodo!

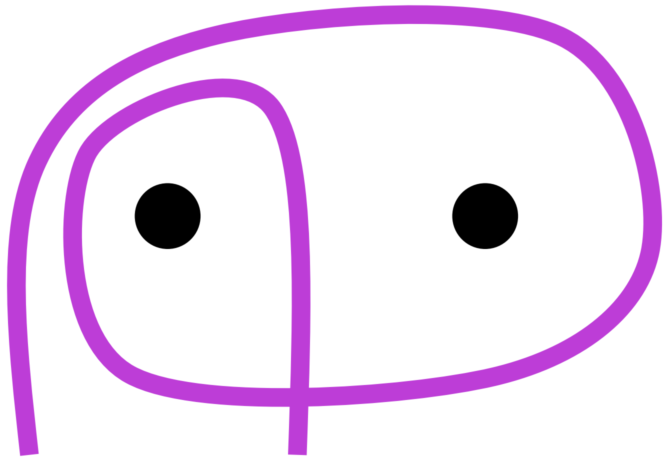
… un modo più perverso.

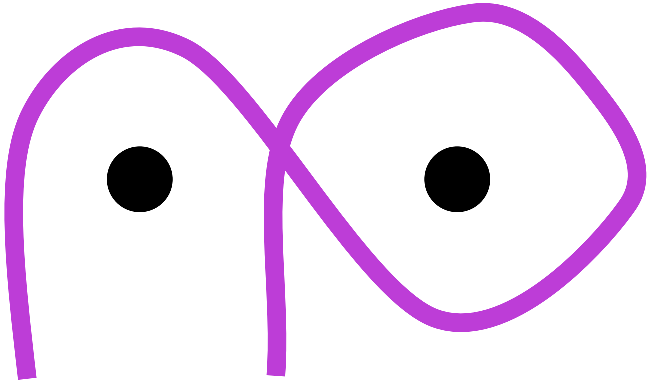
Puzzle (versione base)

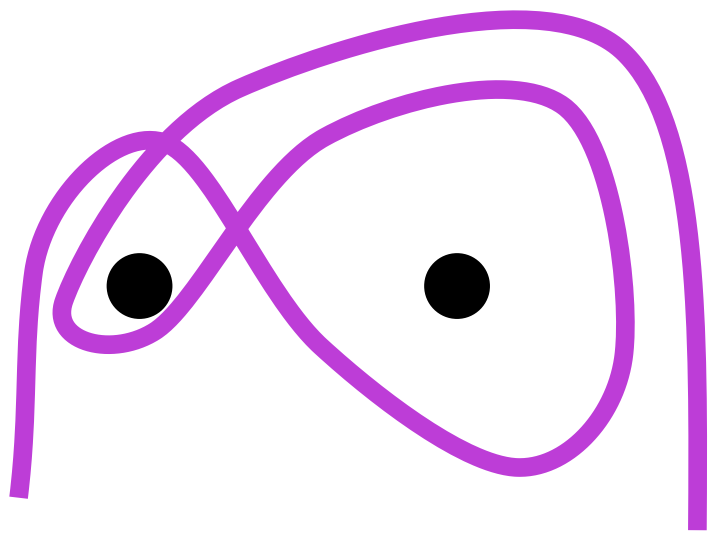
Siano dati due chiodi allineati su un muro, una corda e un quadro. Appendere il quadro al muro arrotolando opportunamente la corda intorno ai chiodi in modo tale che rimuovendo uno qualsiasi dei due chiodi il quadro (per forza di gravità) cada.

...tentativi...

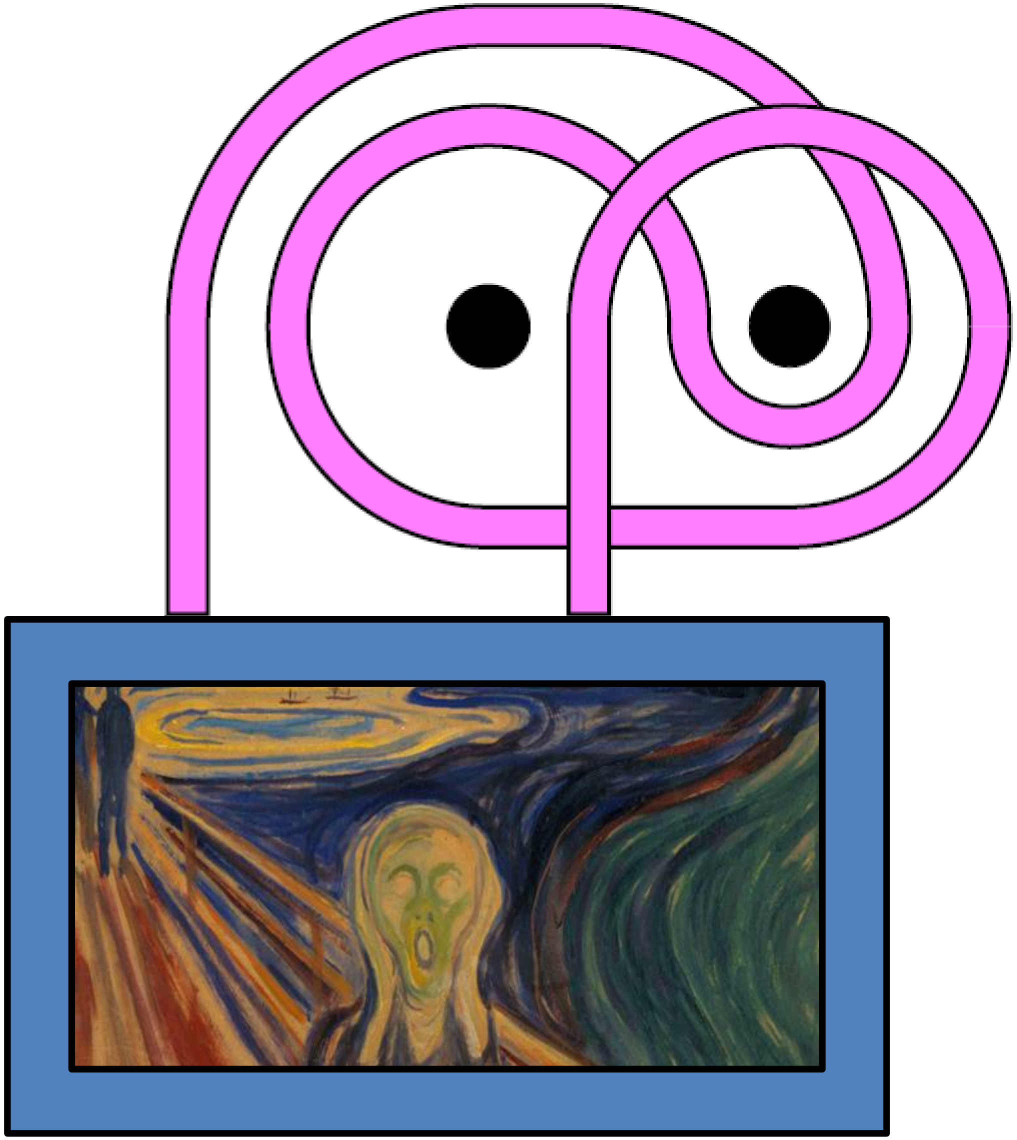








soluzione per due chiodi



adesso se rimuoviamo un chiodo (qualsiasi)?

cade!!!

e se volessi farlo con n chiodi?

n=3,4,...,100,...,1.000.000...

Prima cosa che ho imparato dell'informatica:

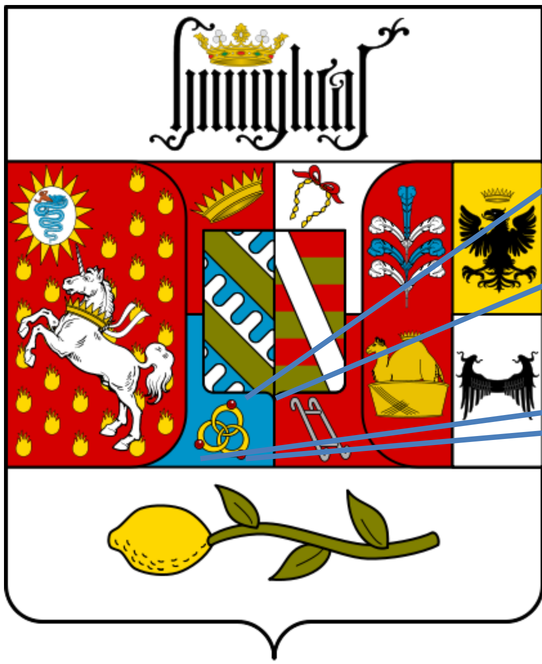
…agli informatici piace pensare in grande.

...ancora più perverso.

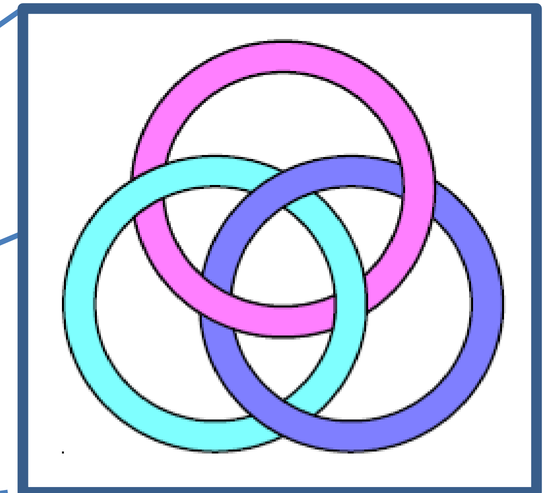
Puzzle (versione più generale)

Siano dati n chiodi allineati su un muro, una corda e un quadro. Appendere il quadro al muro arrotolando opportunamente la corda intorno ai chiodi in modo tale che rimuovendo uno qualsiasi degli n chiodi il quadro (per forza di gravità) cada.

un'interessante relazione: anelli di Borromeo

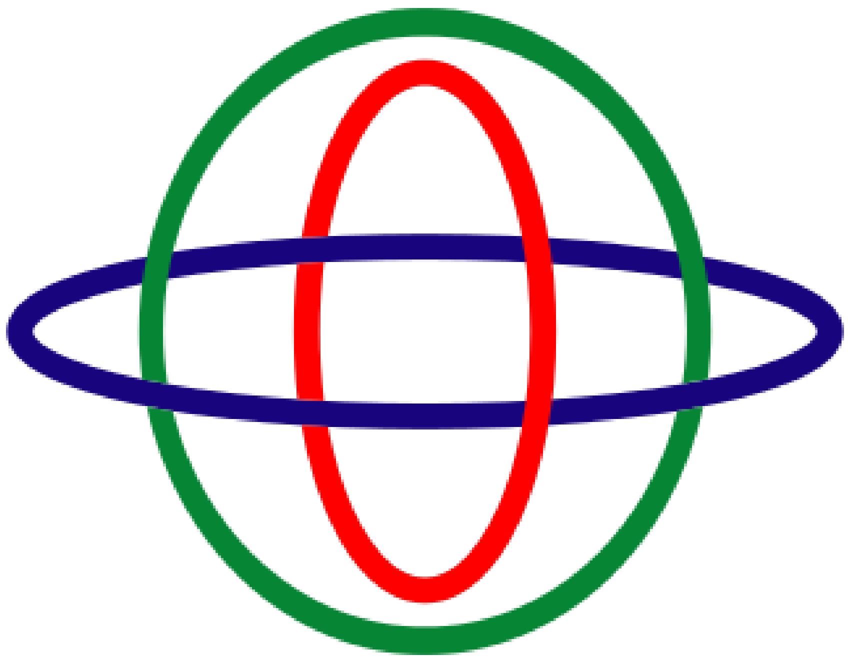


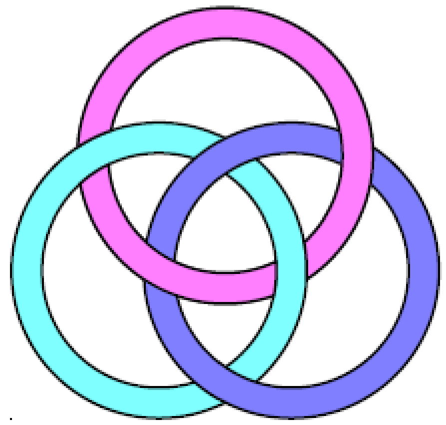
Stemma della famiglia Borromeo, famiglia nobile milanese



tre anelli agganciati, ma rimuovendone uno qualsiasi gli altri due sono liberi

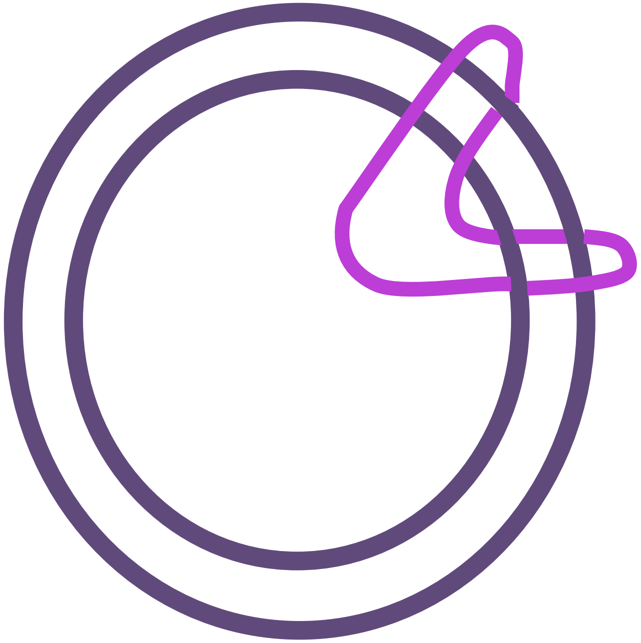
anelli di Borromeo: 3D

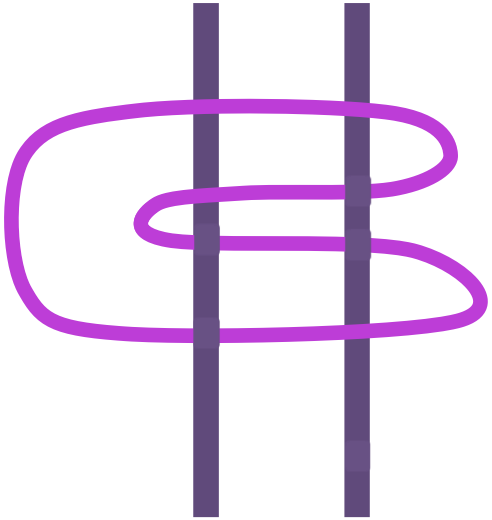




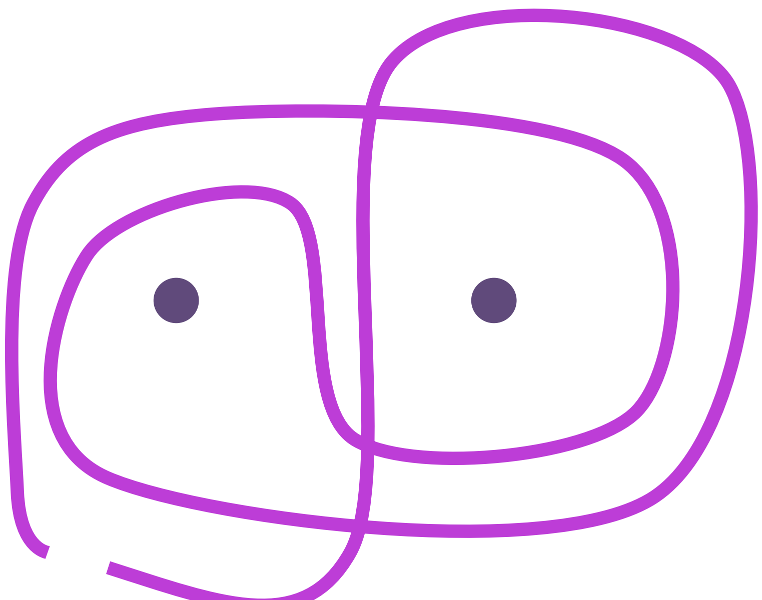
tre anelli agganciati,
ma rimuovendone uno
qualsiasi gli altri due
sono liberi

Un'interessante relazione





anelli di Borromeo:
un altro modo di disegnarli



è la soluzione
del puzzle con due
chiodi!

...torniamo ai quadri.

Puzzle (versione più generale)

Siano dati n chiodi allineati su un muro, una corda e un quadro. Appendere il quadro al muro arrotolando opportunamente la corda intorno ai chiodi in modo tale che rimuovendo uno qualsiasi degli n chiodi il quadro (per forza di gravità) cada.

Il nucleo matematico del problema, ovvero: la formalizzazione

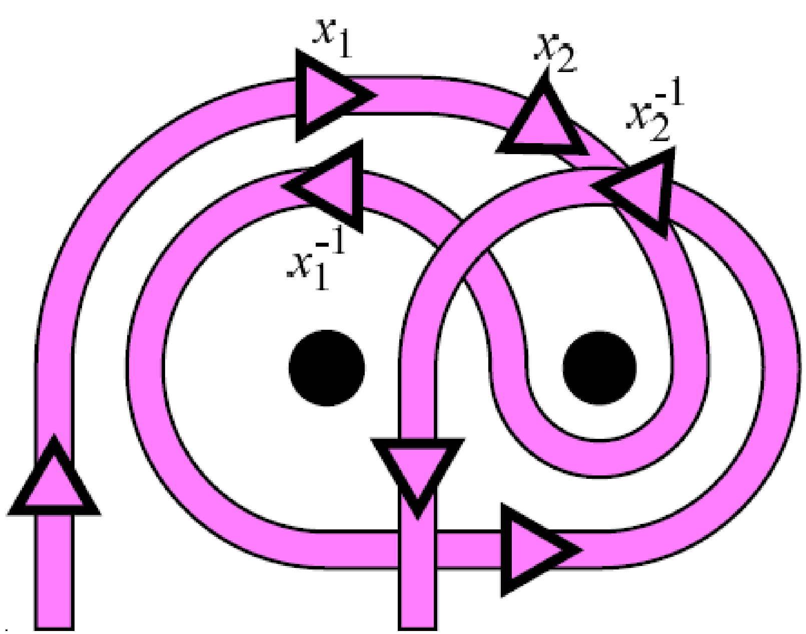
Una astrazione utile che usa i gruppi liberi

2n simboli:

$$x_1, x_1^{-1}, x_2, x_2^{-1}, \; . \; . \; . \; . \; , \; x_n, x_n^{-1}$$

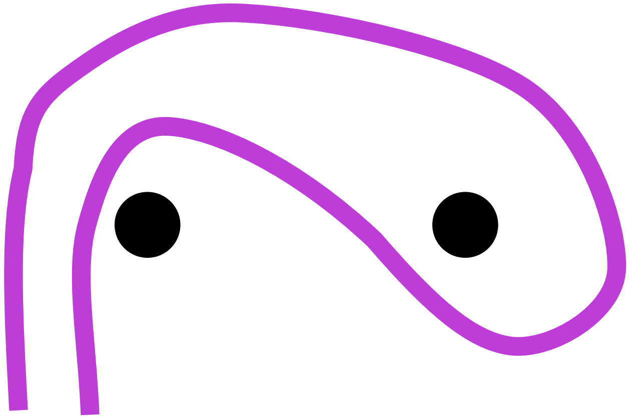
x_i : rappresenta un "giro" intorno al chiodo i in senso orario

x_i^{-1}: rappresenta un "giro" intorno al chiodo i in senso antiorario

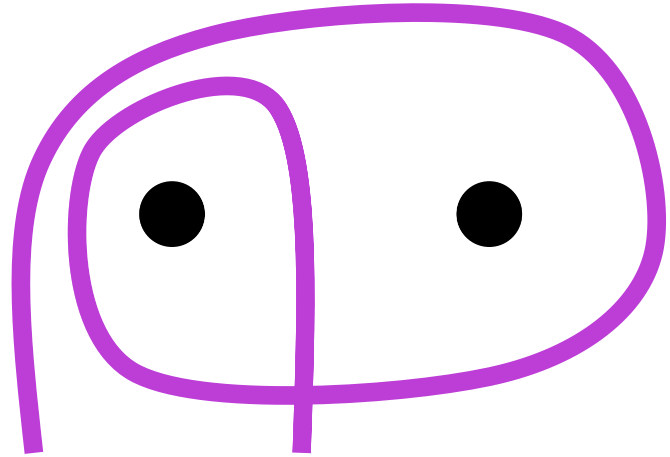


$x_1 \; x_2 \; x_1^{-1} \; x_2^{-1}$

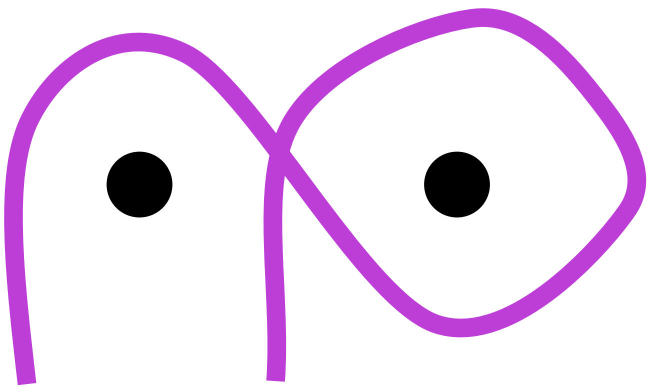
...tentativi...



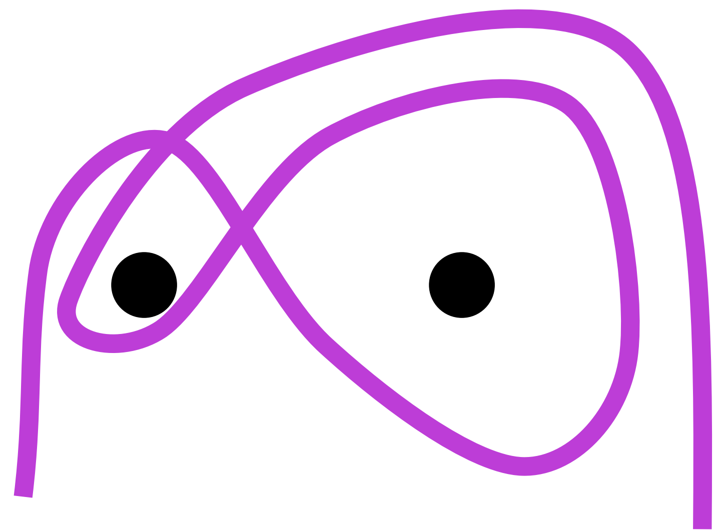
$x_1 \; x_2 \; x_1^{-1}$



$x_1 \; x_2 \; x_1$



$x_1 \; x_2^{-1}$



$x_1 \; x_2^{-1} \; x_1 \; x_2$

Perché formalizzare?

1) per capire proprietà del problema
2) perché una volta formalizzato posso "ragionare" usando la matematica

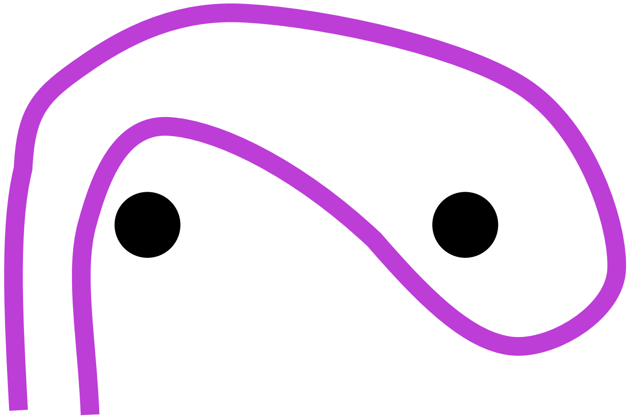
Proprietà

Data un'espressione/arrotolamento, il quadro cade se e solo se l'espressione si cancella.
(e si cancellano solo i termini adiacenti del tipo $x_i\, x_i^{-1}$).

E cosa vuol dire nel modello rimuovere il chiodo i?

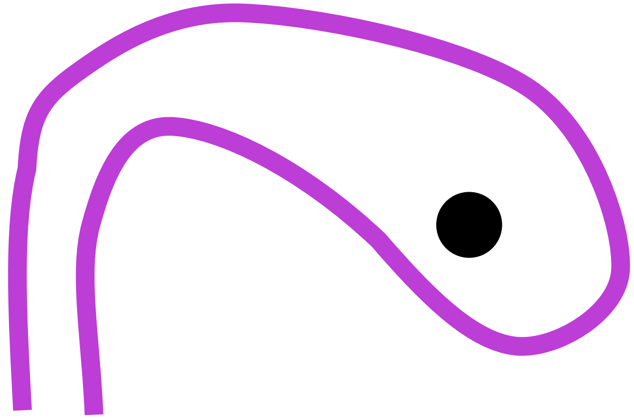
Semplice: cancellare tutte le occorrenze di x_i e x_i^{-1}

...un esempio...



$x_1 \; x_2 \; x_1^{-1}$

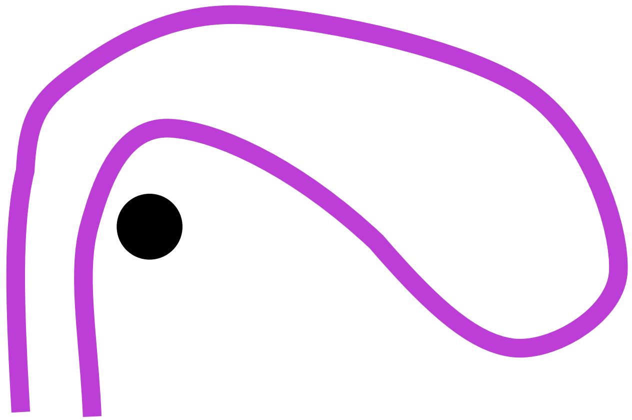
...se rimuovo
primo chiodo...



x_2

non cade!

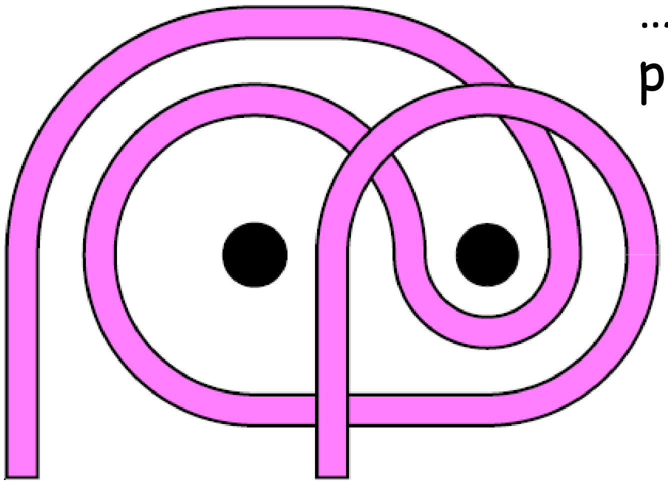
...se rimuovo
secondo chiodo...



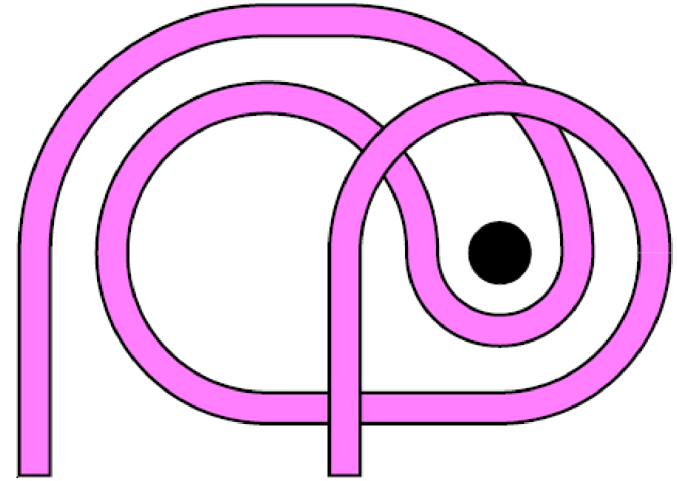
cade!

$\cancel{x_1 \; x_1^{-1}}$

…un altro esempio…



…se rimuovo primo chiodo…

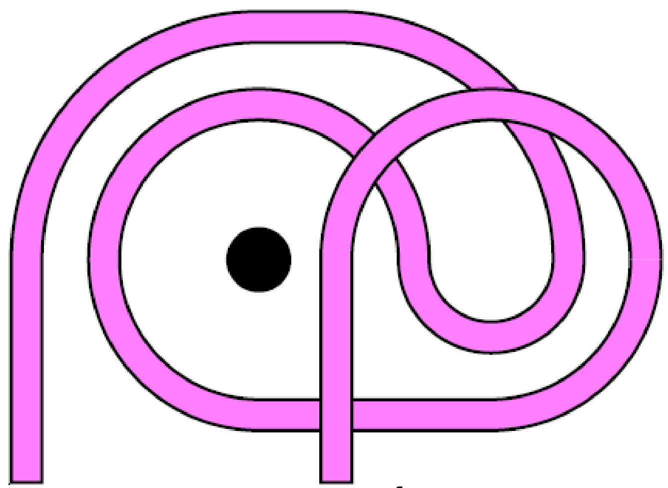


$x_1 \; x_2 \; x_1^{-1} \; x_2^{-1}$

$\cancel{x_2 \; x_2^{-1}}$

cade!

…se rimuovo secondo chiodo…



$\cancel{x_1 \; x_1^{-1}}$

cade!

Dalla formalizzazione all'algoritmo (in questo caso ricorsivo)

Idea: costruire S_n a partire da S_{n-1}.

soluzione per n chiodi: un algoritmo ricorsivo

$$S_2 = \underbrace{x_1 x_2 x_1^{-1} x_2^{-1}}$$

commutatore,
denotato con $[x_1 , x_2]$

Proprietà algebriche:

$(x\, y \ldots z)^{-1} = z^{-1} \ldots y^{-1} x^{-1}$

$(x^{-1})^{-1} = x$

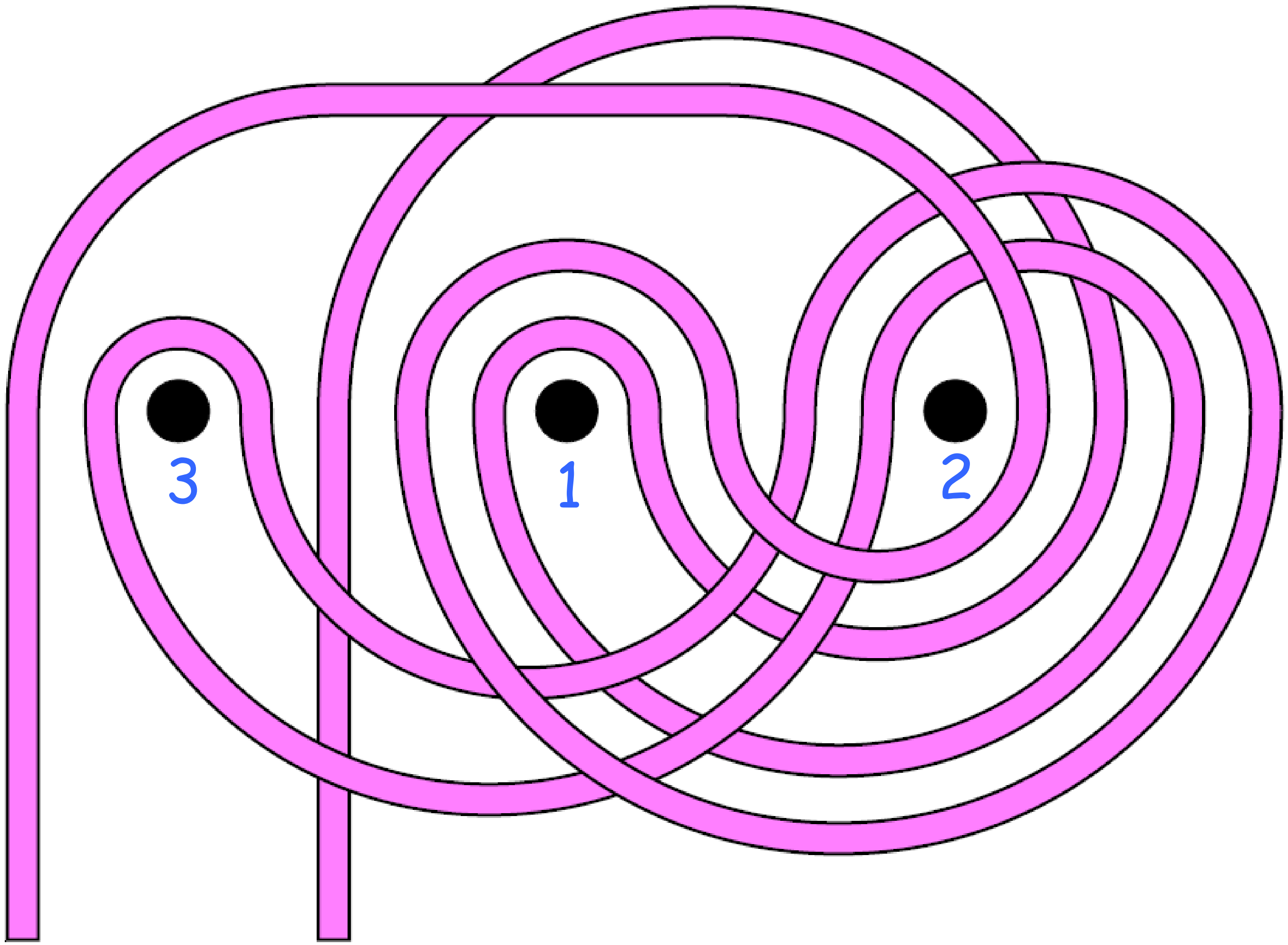
$$S_3 = [\, S_2 , x_3]$$

$$= S_2\, x_3\, S_2^{-1}\, x_3^{-1}$$

$$= x_1 x_2\, x_1^{-1}\, x_2^{-1} x_3 (x_1 x_2\, x_1^{-1}\, x_2^{-1})^{-1} x_3^{-1}$$

$$= x_1 x_2\, x_1^{-1}\, x_2^{-1} x_3 x_2 x_1 x_2^{-1}\, x_1^{-1}\, x_3^{-1}$$

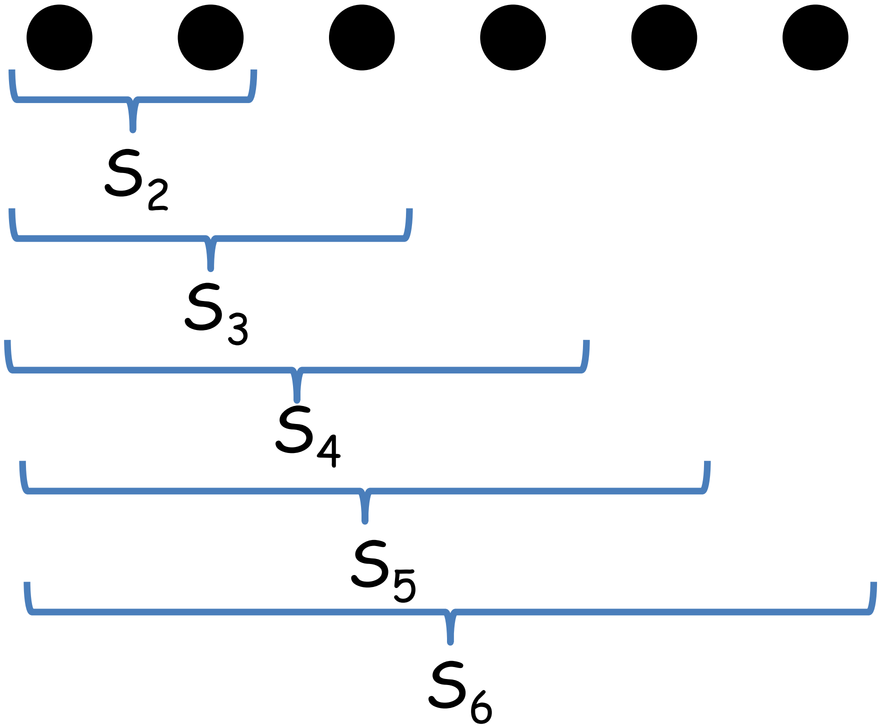
soluzione per tre chiodi



$$x_1 \, x_2 \, x_1^{-1} \, x_2^{-1} \, x_3 \, x_2 \, x_1 \, x_2^{-1} \, x_1^{-1} \, x_3^{-1}$$

Soluzione ricorsiva

$$S_n = [\,S_{n-1}\,,\, x_n\,]$$
$$= S_{n-1}\, x_n\, S_{n-1}^{-1}\, x_n^{-1}$$



Una domanda da informatici:

quanto è buona la soluzione?

quanto serve lunga la corda (in funzione di n)?

quanti simboli ha S_n?

Valutare l'algoritmo, ovvero: analisi della complessità

Analisi della complessità

$$S_n = [\, S_{n-1} \, , \, x_n]$$

$$= S_{n-1}\, x_n\, S_{n-1}^{-1}\, x_n^{-1}$$

$L(n)$: lunghezza (#di simboli) di S_n

$L(n) = 2\,L(n-1) + 2 \Rightarrow \boxed{L(n)=\Theta(2^n)}$

un conto più preciso
(si può dimostrare per induzione)

$$L(n) = 2^n + 2^{n-1} - 2$$

se per ogni simbolo/giro servissero
5 cm, con n=20 chiodi la corda
dovrebbe essere lunga > 78 km!!!

Un'altra cosa che ho imparato dell'informatica:

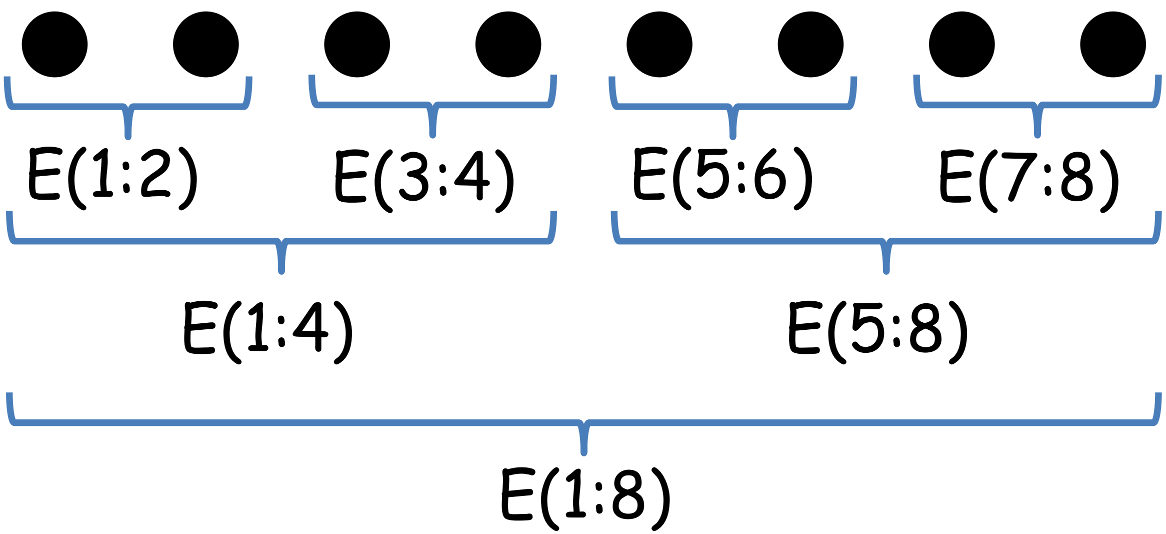
Se un problema lo risolvi male, è come se non l'hai risolto per niente.

L'eterno tarlo dell'algoritmista: si potrà fare meglio?

Idea: costruire S_n in modo più "bilanciato", in termini di $S_{n/2}$ e non di S_{n-1}.

Una soluzione più efficiente

$E(i:j)$: soluzione per i chiodi da i a j



$$E(i:i) = x_i$$

$$E(i:i+1) = [x_i\,,\, x_{i+1}] = x_i\, x_{i+1}\, x_i^{-1}\, x_{i+1}^{-1}$$

$$E(i:j) = \left[E\left(i : \left\lfloor (i+j)/2 \right\rfloor\right), E\left(\left\lfloor (i+j)/2 \right\rfloor + 1 : j\right)\right]$$

Analisi della complessità

$E(i:j)$: soluzione per i chiodi da i a j

$$E(i:i) = x_i$$

$$E(i:i+1) = [x_i\ , x_{i+1}] = x_i\ x_{i+1}\ x_i^{-1}\ x_{i+1}^{-1}$$

$$E(i:j) = \left[E(i:\lfloor (i+j)/2 \rfloor),\ E(\lfloor (i+j)/2 \rfloor +1 : j)\right]$$

$L(n)$: lunghezza (#di simboli) di S_n

corda eponenzialemnte più corta!

$L(n) = 4\,L(n/2)$

$L(1) = 1$
$L(2) = 4$

$$\boxed{L(n)=\Theta(n^2)}$$

Le due soluzioni a confronto

$L(n)$: lunghezza (#di simboli) di S_n

prima soluzione

$L(2) = 4$

$L(4) = 22$

$L(8) = 382$

$L(16) = 98.302$

$$L(n) \approx 2^n$$

se per ogni simbolo/giro servissero 5 cm, con n=20 chiodi la corda dovrebbe essere lunga > 78 km!!!

seconda soluzione

$L(2) = 4$

$L(4) = 16$

$L(8) = 64$

$L(16) = 256$

$$L(n) \approx n^2$$

con n=20 chiodi serve un corda di circa 20 metri!


…THE END…

soluzione per quattro chiodi

$$x_1 x_2 x_1^{-1} x_2^{-1} x_3 x_4 x_3^{-1} x_4^{-1} x_2 x_1 x_2^{-1} x_1^{-1} x_4 x_3 x_4^{-1} x_3^{-1}$$