Il tempo è una risorsa anche per un computer: Un pizzico di “Computational Thinking”

Claudio Mirolo

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

claudio.mirolo@uniud.it

PLS di Informatica

nid.dimi.uniud.it

Sommario

Percorso

1. Formiche e computer
   - gedankenexperiment
   - analisi
   - modello
2. Torre di Hanoi
   - leggenda
   - varianti
3. Problema delle N regine
   - ricerca esaustiva
4. Epilogo

Gedankenexperiment: Formica sul nastro elastico. . .

Gedankenexperiment: Formica sul nastro elastico. . .

Formichina sul nastro elastico. . .

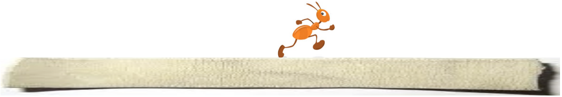
Formichina sul nastro elastico. . .

Formichina sul nastro elastico. . .

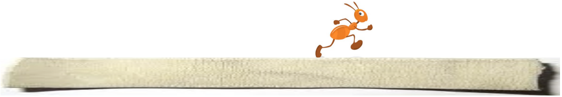
Formichina sul nastro elastico. . .

Formichina sul nastro elastico. . .

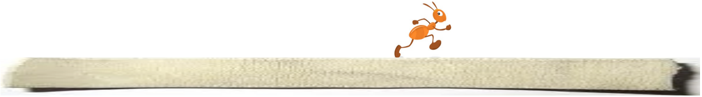
Formichina sul nastro elastico. . .



Formichina sul nastro elastico. . .



Formichina sul nastro elastico. . .



Formichina sul nastro elastico. . .

Formichina sul nastro elastico. . .

Formichina sul nastro elastico. . .

Formichina sul nastro elastico. . .

Formichina sul nastro elastico...

Ce la farà a raggiungere l’altro capo?

Problemino

- La formichina percorre 1 *cm* in 10^{-1} *sec*
- Il nastro elastico a riposo misura 4 *cm*
- Ad ogni avanzamento di 1 *cm* della formichina corrisponde un allungamento di 4 *cm* del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

Problemino

- La formichina percorre 1 *cm* in 10^{-1} *sec*
- Il nastro elastico a riposo misura 4 *cm*
- Ad ogni avanzamento di 1 *cm* della formichina corrisponde un allungamento di 4 *cm* del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

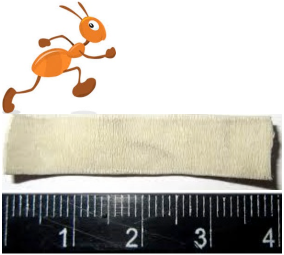
Problemino

- La formichina percorre 1 cm in 10^{-1} sec
- Il nastro elastico a riposo misura 4 cm
- Ad ogni avanzamento di 1 cm della formichina corrisponde un allungamento di 4 cm del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

Problemino

- La formichina percorre 1 *cm* in 10^{-1} *sec*
- Il nastro elastico a riposo misura 4 *cm*
- Ad ogni avanzamento di 1 *cm* della formichina corrisponde un allungamento di 4 *cm* del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

analisi



1 cm

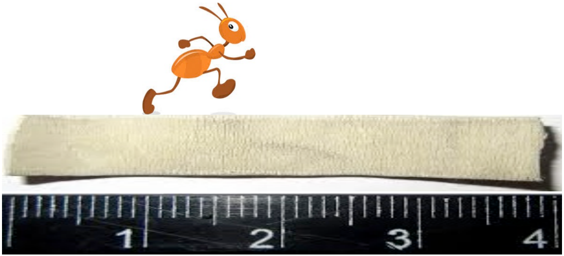
analisi

1 cm

analisi

1 cm

Analisi: Proporzioni. . .



$1 \quad + \quad \frac{1}{2}$ cm stirato

Analisi: Proporzioni. . .

$1 \quad + \quad \frac{1}{2}$ cm stirato

Analisi: Proporzioni. . .

$1 \quad + \quad \frac{1}{2}$ cm stirato

Analisi: Proporzioni. . .

$1 \quad + \quad \frac{1}{2} \quad + \quad \frac{1}{3}$ cm stirato

Analisi: Proporzioni. . .

$1 \quad + \quad \frac{1}{2} \quad + \quad \frac{1}{3}$ cm stirato

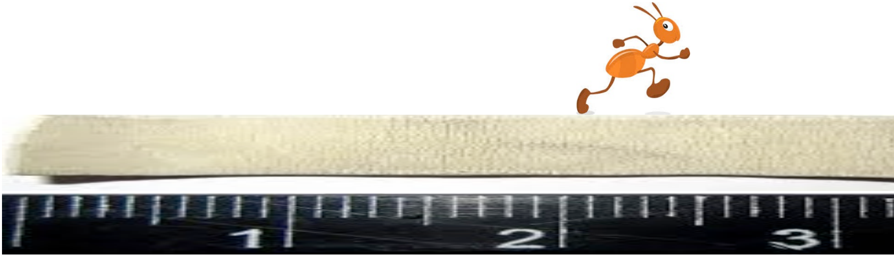
Analisi: Proporzioni. . .

$1 \quad + \quad \frac{1}{2} \qquad + \quad \frac{1}{3}$ cm stirato

Analisi: Proporzioni. . .

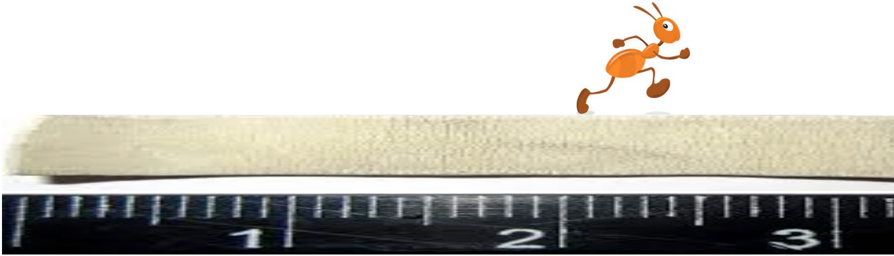
$1 \quad + \quad \frac{1}{2} \quad + \quad \frac{1}{3} \quad + \quad \frac{1}{4}$ cm stirato

Analisi: Proporzioni. . .



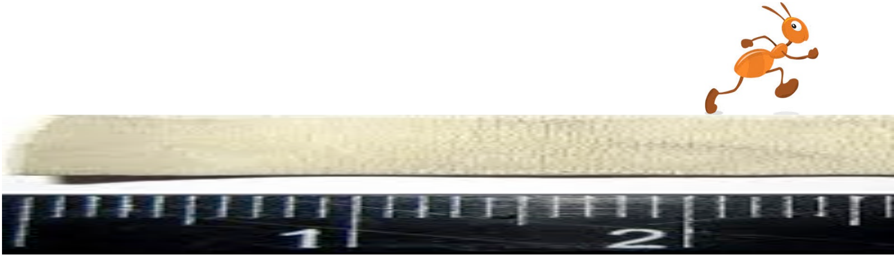
$1 \quad + \quad \frac{1}{2} \quad + \quad \frac{1}{3} \quad + \quad \frac{1}{4}$ cm stirato

Analisi: Proporzioni. . .



$$1 \quad + \quad \frac{1}{2} \quad + \quad \frac{1}{3} \quad + \quad \frac{1}{4}$$ cm stirato

Analisi: Proporzioni. . .



$$1 \quad + \frac{1}{2} \quad + \frac{1}{3} \quad + \frac{1}{4} \quad \ldots$$

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati
- Risposta?

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati
- Risposta?

La formichina ce la farà?

- $$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots \qquad \text{cm stirati}$$

- Risposta?

$$1 \quad = \quad 1.00 \text{ cm stirati}$$

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati

- Risposta?

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \quad = \quad 2.08 \text{ cm stirati}$$

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati
- Risposta?

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} = 2.83 \text{ cm stirati}$$

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati

- Risposta?

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9}$$

$$+ \frac{1}{10} + \frac{1}{11} \quad = \quad 3.02 \text{ cm stirati}$$

La formichina ce la farà?

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \ldots$ cm stirati
- Risposta?

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9}$$

$$+ \frac{1}{10} + \frac{1}{11} + \frac{1}{12} + \frac{1}{13} + \ldots + \frac{1}{31} = 4.03 \text{ cm stirati}$$

Risposta di un matematico

- La formichina percorre 1 *cm* in 10^{-1} *sec*
- Il nastro elastico a riposo misura 1 *m*
- Ad ogni avanzamento di 1 *cm* della formichina corrisponde un allungamento di 1 *m* del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

Problemino rivisitato. . .

- La formichina percorre 1 *cm* in 10^{-1} *sec*
- Il nastro elastico a riposo misura 1 *m*
- Ad ogni avanzamento di 1 *cm* della formichina corrisponde un allungamento di 1 *m* del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

Problemino rivisitato. . .

- La formichina percorre 1 cm in 10^{-1} sec
- Il nastro elastico a riposo misura 1 m
- Ad ogni avanzamento di 1 cm della formichina corrisponde un allungamento di 1 m del nastro
- Riuscirà la formica a spostarsi da un estremo all'altro?

Conclusioni di un matematico

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$ diverge!

- Esiste k (cm effettivi) tale che
  $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k} \geq n$ (cm stirati)

- La formichina è in grado di percorrere tutto il nastro (qualunque sia la lunghezza)

Conclusioni di un matematico: più in generale...

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$ diverge!

- Esiste k (cm effettivi) tale che
  $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{k} \geq n$ (cm stirati)

- La formichina è in grado di percorrere tutto il nastro
  (qualunque sia la lunghezza)

Conclusioni di un matematico: più in generale...

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$ diverge!
- Esiste k (cm effettivi) tale che
  $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k} \geq n$ (cm stirati)
- La formichina è in grado di percorrere tutto il nastro (qualunque sia la lunghezza)

Conclusioni di un matematico: più in generale...

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$ diverge!

- Esiste k (cm effettivi) tale che
  $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k} \geq n$ (cm stirati)

- La formichina è in grado di percorrere tutto il nastro
  (qualunque sia la lunghezza)

Conclusioni di un matematico: più in generale...

- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \dots$ diverge!
- Esiste k (cm effettivi) tale che
  $H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k} \geq n$ (cm stirati)
- La formichina è in grado di percorrere tutto il nastro (qualunque sia la lunghezza)

Un modello (informatico)

```
public static long antJogging( int n ) {

  long k = 1;
  double h = 1.0;

  while ( h < n ) {

    k = k + 1;
    h = h + 1.0 / k;
  }
  return k;
}
```

La formichina è il computer che esegue questo algoritmo:

```
public static long antJogging( int n ) {

  long k = 1;
  double h = 1.0;

  while ( h < n ) {

    k = k + 1;
    h = h + 1.0 / k;
  }
  return k;
}
```

Osservazioni

- `...  antJogging( 4 )  ...  ?`
- `...  antJogging( 100 )  ...  ?`
- Che relazione c'è fra i tempi di calcolo di `antJogging(` $n+1$ `)` e `antJogging(` n `)`

  AntRide.java
- Quanto impiega la formica per coprire 1 cm stirato in più?

Osservazioni

- ... `antJogging( 4 )` ... ?
- ... `antJogging( 100 )` ... ?
- Che relazione c'è fra i tempi di calcolo di `antJogging(` $n+1$ `)` e `antJogging(` n `)`

  AntRide.java

- Quanto impiega la formica per coprire 1 cm stirato in più?

Osservazioni

- `... antJogging( 4 ) ... ?`
- `... antJogging( 100 ) ... ?`
- Che relazione c'è fra i tempi di calcolo di
  `antJogging(` $n+1$ `)` e `antJogging(` n `)`

  AntRide.java
- Quanto impiega la formica per coprire 1 cm stirato in più?

Osservazioni

- `... antJogging( 4 ) ... ?`
- `... antJogging( 100 ) ... ?`
- Che relazione c'è fra i tempi di calcolo di

  antJogging($n+1$) e antJogging(n)

  AntRide.java
- Quanto impiega la formica per coprire 1 cm stirato in più?

Osservazioni

- 4 *cm* in $31 \cdot 10^{-1}$ *sec* $=$ 3.1 *sec*
- 5 *cm* in $83 \cdot 10^{-1}$ *sec* $=$ 8.3 *sec*
- 100 *cm* in (molto) più di $8.3 \cdot 2 \cdot 2 \cdot 2 \cdot \ \ldots$ *sec*

Osservazioni

- $4\ cm$ in $31 \cdot 10^{-1}\ sec = 3.1\ sec$
- $5\ cm$ in $83 \cdot 10^{-1}\ sec = 8.3\ sec$
- $100\ cm$ in (molto) più di $8.3 \cdot 2 \cdot 2 \cdot 2 \cdot\ \ldots\ sec$

Osservazioni

- $4\ cm$ in $31 \cdot 10^{-1}\ sec = 3.1\ sec$
- $5\ cm$ in $83 \cdot 10^{-1}\ sec = 8.3\ sec$
- $100\ cm$ in (molto) più di $8.3 \cdot 2 \cdot 2 \cdot 2 \cdot \ \ldots\ sec$

Risposta di uno scienziato

- tempo
- A quanti anni corrisponde?
- A quanti miliardi di anni?
- Qual è la vita stimata dell'universo?

Conclusioni di uno scienziato (Fisico?)

- tempo ?
- A quanti anni corrisponde?
- A quanti miliardi di anni?
- Qual è la vita stimata dell'universo?

Conclusioni di uno scienziato (Fisico?)

- tempo $\gg$ $3.3 \cdot 10^{29}$ *sec*
- A quanti anni corrisponde?
- A quanti miliardi di anni?
- Qual è la vita stimata dell'universo?

Conclusioni di uno scienziato (Fisico?)

- tempo $\gg$ $3.3 \cdot 10^{29}$ *sec*
- A quanti anni corrisponde?
- A quanti miliardi di anni?
- Qual è la vita stimata dell'universo?

Conclusioni di uno scienziato (Fisico?)

- tempo $\gg$ $3.3 \cdot 10^{29}$ *sec*
- A quanti anni corrisponde?
- A quanti miliardi di anni?
- Qual è la vita stimata dell'universo?

Tirando le somme...

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo...
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta:
- ...

Tirando le somme...

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo...
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta:
- ...

Tirando le somme. . .

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo. . .
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta:
- . . .

Tirando le somme. . .

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo. . .
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta:
- . . .

Tirando le somme...

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo...
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta: ?
- ...

Tirando le somme. . .

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo. . .
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta: (molto) più di 1000 miliardi di anni
- . . .

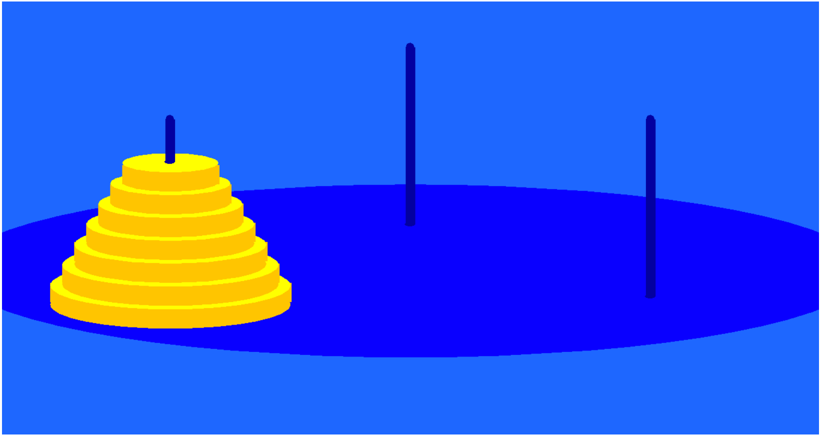
Tirando le somme. . .

- Ogni spostamento di 1 cm della formica corrisponde a un'iterazione del programma
- Ma un computer è velocissimo. . .
- E quanto impiegherebbe, allora, un computer? (supponiamo: una somma in 10^{-10} *sec*)
- Risposta: (molto) più di 1000 miliardi di anni
- . . . Eppure sembrava un programma così innocuo!

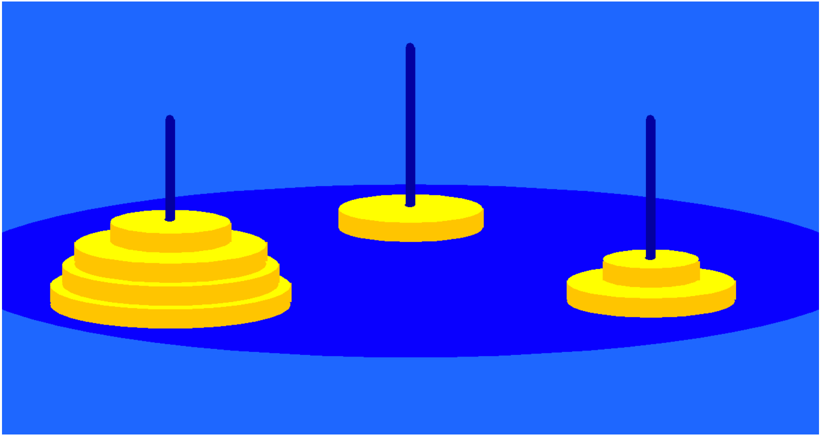
Percorso

1. Formiche e computer
   - gedankenexperiment
   - analisi
   - modello
2. Torre di Hanoi
   - leggenda
   - varianti
3. Problema delle N regine
   - ricerca esaustiva
4. Epilogo

Torre di Hanoi



Torre di Hanoi



Semplice programma ricorsivo

```
public static void hanoi( int n,
                          int s, int d, int t ) {

  if ( n > 0 ) {

    hanoi( n-1,  s, t, d );
    move( s, d );
    hanoi( n-1,  t, d, s );
  }
}
```

Osservazioni

- `hanoi( 1, ... ) ... quante mosse?`
- `hanoi( 2, ... ) ... quante mosse?`
- `hanoi( 3, ... ) ... quante mosse?`
- `hanoi( 4, ... ) ... quante mosse?`
- Quante mosse per una torre di n dischi?

hanoi.jar

Osservazioni

- `hanoi( 1, ... ) ... quante mosse?`
- `hanoi( 2, ... ) ... quante mosse?`
- `hanoi( 3, ... ) ... quante mosse?`
- `hanoi( 4, ... ) ... quante mosse?`
- Quante mosse per una torre di n dischi?

hanoi.jar

Osservazioni

- `hanoi( 1, ... )   ...  quante mosse?`
- `hanoi( 2, ... )   ...  quante mosse?`
- `hanoi( 3, ... )   ...  quante mosse?`
- `hanoi( 4, ... )   ...  quante mosse?`
- Quante mosse per una torre di n dischi?

hanoi.jar

Osservazioni

- `hanoi( 1, ... )   ...  quante mosse?`
- `hanoi( 2, ... )   ...  quante mosse?`
- `hanoi( 3, ... )   ...  quante mosse?`
- `hanoi( 4, ... )   ...  quante mosse?`
- Quante mosse per una torre di n dischi?

hanoi.jar

Osservazioni

- `hanoi( 1, ... )   ...  quante mosse?`
- `hanoi( 2, ... )   ...  quante mosse?`
- `hanoi( 3, ... )   ...  quante mosse?`
- `hanoi( 4, ... )   ...  quante mosse?`
- Quante mosse per una torre di n dischi?

hanoi.jar

Torre di Hanoi

Il matematico francese *Édouard Lucas* inventò questo gioco nel 1883 e lo commercializzò accompagnandolo con una leggenda che narra di un monaco Indù alle prese con il rompicapo per una torre di 64 dischi d'oro e di una profezia che nel momento in cui avrà completato tutte le mosse il mondo finirà.

In effetti il numero di mosse per spostare una torre di 64 dischi rispettando le regole stabilite è enorme, al punto che non basterebbe il tempo di vita dell'universo (secondo le stime attuali) per portare a termine il compito.

Torre di Hanoi

Il matematico francese *Édouard Lucas* inventò questo gioco nel 1883 e lo commercializzò accompagnandolo con una leggenda che narra di un monaco Indù alle prese con il rompicapo per una torre di 64 dischi d'oro e di una profezia che nel momento in cui avrà completato tutte le mosse il mondo finirà.

In effetti il numero di mosse per spostare una torre di 64 dischi rispettando le regole stabilite è enorme, al punto che non basterebbe il tempo di vita dell'universo (secondo le stime attuali) per portare a termine il compito.

Quanto tempo impiegherebbe il monaco?

- $2^{64} - 1$ mosse $\times$ 1 sec/mossa $=$ $\ldots ?$
- $>$ 500 miliardi di anni!

Quanto tempo impiegherebbe il monaco?

- $2^{64} - 1$ mosse $\times$ 1 sec/mossa $=$ $\ldots$?
- $>$ 500 miliardi di anni!

E se ci interessasse conoscere una sola mossa?

- k-ima mossa relativa a una torre di n dischi:

- `hMove( n, k, ... )   ...  quale mossa?`

- Per esempio per $n = 30$ e $k = 500\,000\,000$

E se ci interessasse conoscere una sola mossa?

- k-ima mossa relativa a una torre di n dischi:
- `hMove( n, k, ... )   ...   quale mossa?`
- Per esempio per $n = 30$ e $k = 500\,000\,000$

E se ci interessasse conoscere una sola mossa?

- k-ima mossa relativa a una torre di n dischi:
- `hMove( n, k, ... )   ...  quale mossa?`
- Per esempio per $n = 30$ e $k = 500\,000\,000$

La soluzione che viene in mente per prima. . .

```
public static void hMove( int n, long k,
                          int s, int d, int t ) {

  if ( n > 0 ) {
    hMove( n-1, k,  s, t, d );
    if ( count < k ) {
      count = count + 1;
      if ( count == k ) {
        move( s, d );
      } else {
        hMove( n-1, k,  t, d, s );
    }}
  }
}
```

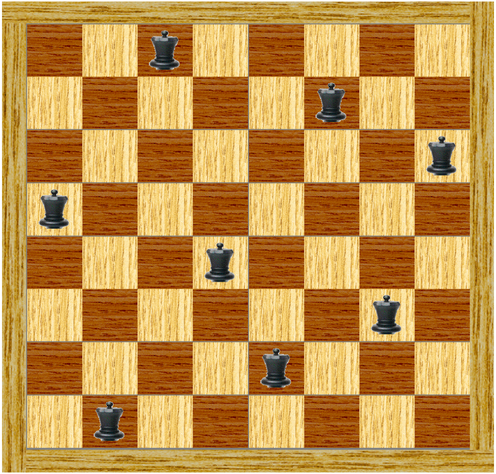
. . . ma si può fare molto meglio

```
public static void qMove( int n, long k,
                          int s, int d, int t ) {

  long m = (long) pow2( n-1 );

  if ( k < m ) {
    qMove( n-1, k,   s, t, d );
  } else if ( k == m ) {
    move( s, d );
  } else {
    qMove( n-1, k-m,  t, d, s );
  }
}
```

Percorso

1. Formiche e computer
   - gedankenexperiment
   - analisi
   - modello
2. Torre di Hanoi
   - leggenda
   - varianti
3. Problema delle N regine
   - ricerca esaustiva
4. Epilogo

Problema delle N regine



Quante soluzioni per una scacchiera $n \times n$?

- Il problema delle *N regine*
  è stato posto verso la metà dell' '800
- Anche per determinare il numero di soluzioni
  non si conoscono algoritmi migliori
  di quelli basati su una ricerca sistematica. . .
- Tant'è che se ne conosce il numero preciso
  solo fino a $n = 27$ $(> 2 \cdot 10^{17})$

Quante soluzioni per una scacchiera $n \times n$?

- Il problema delle *N regine*
  è stato posto verso la metà dell' '800
- Anche per determinare il numero di soluzioni
  non si conoscono algoritmi migliori
  di quelli basati su una ricerca sistematica...
- Tant'è che se ne conosce il numero preciso
  solo fino a $n = 27$ $\quad (> 2 \cdot 10^{17})$

Quante soluzioni per una scacchiera $n \times n$?

- Il problema delle *N regine*
  è stato posto verso la metà dell' '800

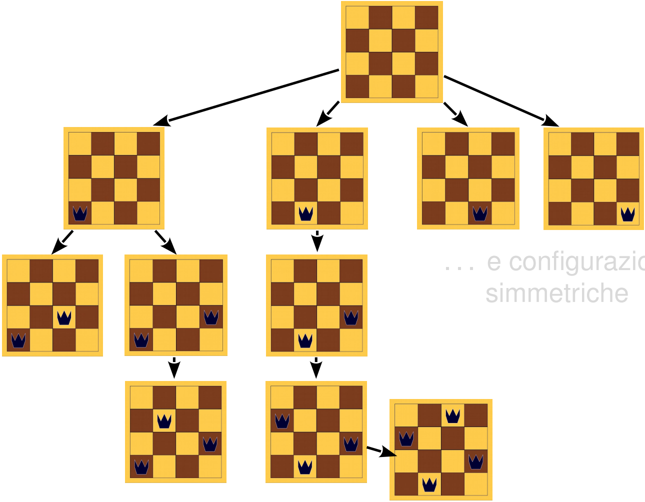
- Anche per determinare il numero di soluzioni
  non si conoscono algoritmi migliori
  di quelli basati su una ricerca sistematica...

- Tant'è che se ne conosce il numero preciso
  solo fino a $n = 27 \quad (> 2 \cdot 10^{17})$

Quante soluzioni per una scacchiera $n \times n$?

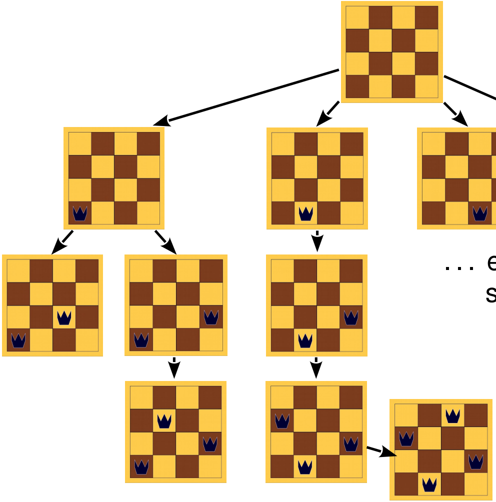
n	1	2	3	4	5	6	7	8	9	10
soluzioni	1	0	0	2	10	4	40	92	352	724

Enumerazione esaustiva delle soluzioni



… e configurazioni simmetriche

Enumerazione esaustiva delle soluzioni



. . . e configurazioni simmetriche

Contare le soluzioni. . .

- Per $n = 27$ il numero di soluzioni è $> 2 \cdot 10^{17}$
- Solo per contare le soluzioni è necessario sommare 1 più di $2 \cdot 10^{17}$ volte
- Quanto tempo impiegherebbe un computer?

Contare le soluzioni. . .

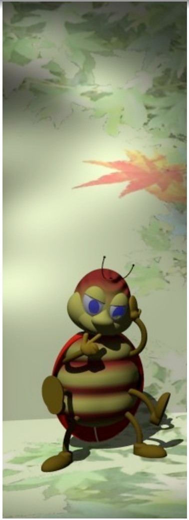
- Per $n = 27$ il numero di soluzioni è $> 2 \cdot 10^{17}$
- Solo per contare le soluzioni è necessario sommare 1 più di $2 \cdot 10^{17}$ volte
- Quanto tempo impiegherebbe un computer?

Contare le soluzioni. . .

- Per $n = 27$ il numero di soluzioni è $> 2 \cdot 10^{17}$
- Solo per contare le soluzioni è necessario sommare 1 più di $2 \cdot 10^{17}$ volte
- Quanto tempo impiegherebbe un computer?

Percorso

1. Formiche e computer
   - gedankenexperiment
   - analisi
   - modello
2. Torre di Hanoi
   - leggenda
   - varianti
3. Problema delle N regine
   - ricerca esaustiva
4. Epilogo



Tempo di calcolo

- Per quanto il computer sia veloce...
- Il tempo di elaborazione è una risorsa...
- ...che non sempre si può trascurare!

Tempo di calcolo

- Per quanto il computer sia veloce. . .
- Il tempo di elaborazione è una risorsa. . .
- . . . che non sempre si può trascurare!

Tempo di calcolo

- Per quanto il computer sia veloce...
- Il tempo di elaborazione è una risorsa...
- ...che non sempre si può trascurare!

The End

Domande?

The End

Grazie