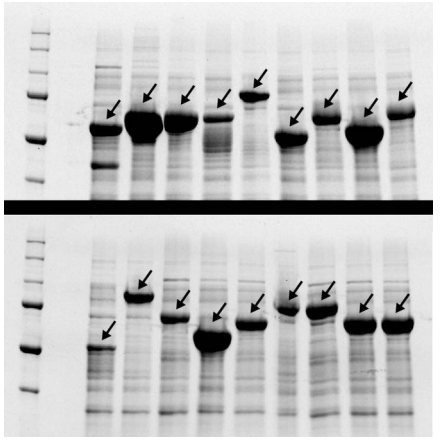
DNA computing:
come usare il DNA per risolvere problemi irrisolvibili con computer normali

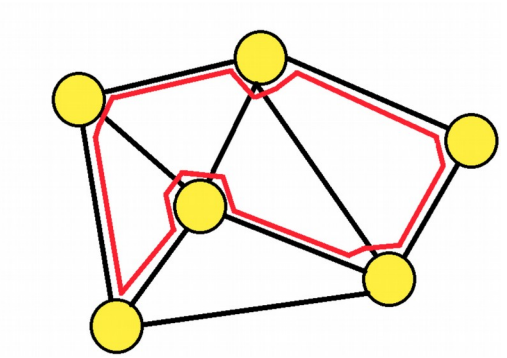


Struttura presentazione

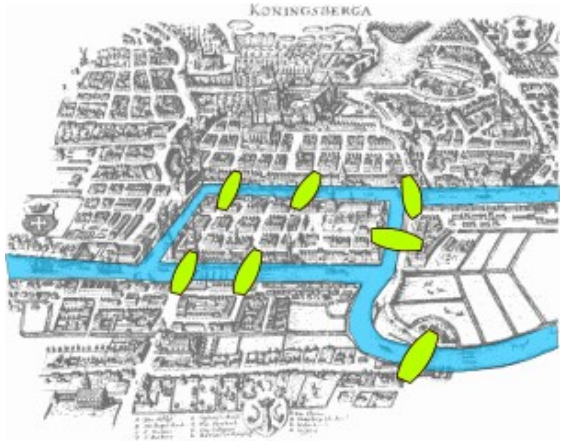
- Quali sono i problemi irrisovibili dai computer classici
  (problemi NP completi)
- Come possono influire sulla nostra vita questi problemi
  (sicurezza in Internet)
- Come puo' un computer non classico (a DNA) risolvere questi problemi
  (con che mezzi ed esperimenti)
- Esperimento di Adleman per dimostare la risolvibilita' di un problema NP-completo con computazione mediante DNA
- Confronto reale tra computer convenzionali e non convenzionali

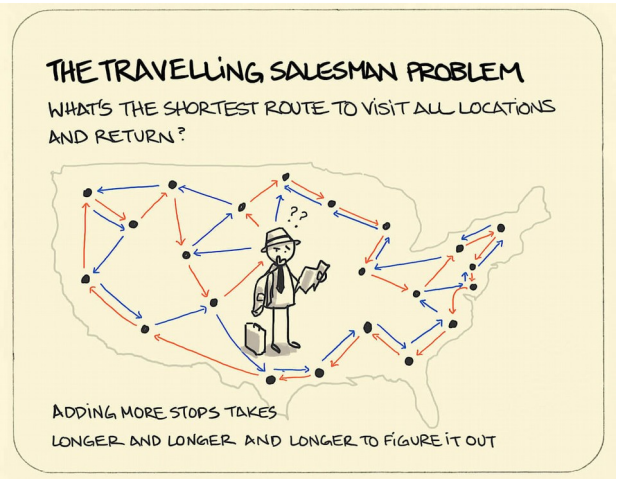
Problemi NP-Completi

- Tipo di problemi molto difficili per un calcolatore perche’ non si trovano algoritmi veloci
- Il tempo di risoluzione cresce spesso in modo esponenziale all’aumentare della grandezza del problema
- Sulla loro irrisolvibilita’ basiamo cose molto pratiche, per esempio la sicurezza in Internet e delle comunicazioni in generale



“7 ponti di Konigsberg” - “Problema del commesso viaggiatore” entrambi problemi NP-Hard (come decifrare RSA)

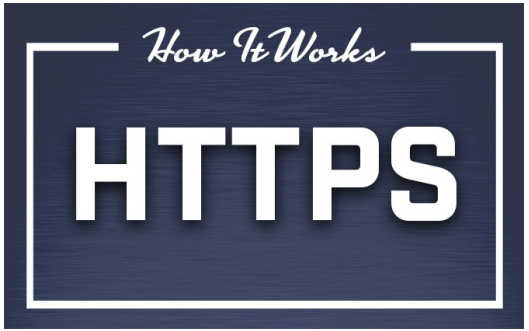




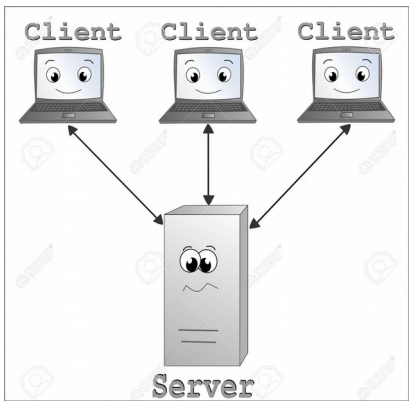
- “Esiste un percorso che ritorni al punto di partenza passando solo una volta per ciascun ponte (citta’)?”
- “Se esistono piu’ percorsi, qual’e’ il piu’ breve?”
- Un computer “normale” deve provare tutte le possibilita’ una alla volta (“brute force”)

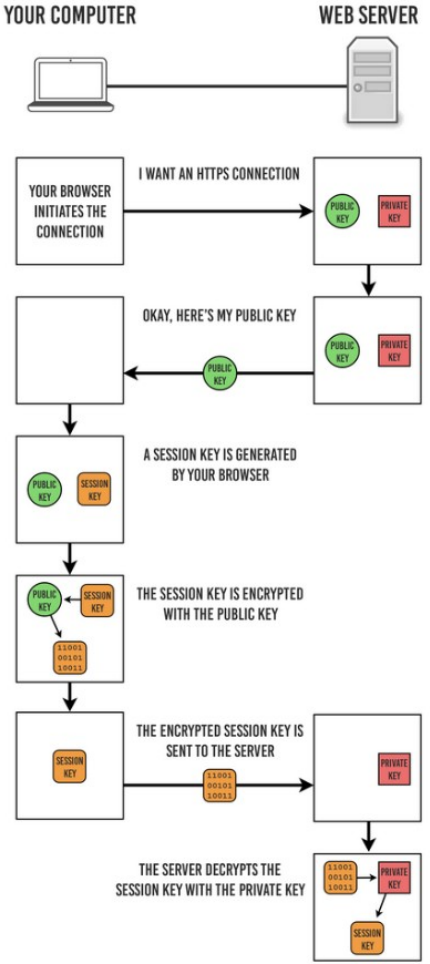
Dati Internet - RSA Encryption

- In crittografia la sigla RSA indica un algoritmo di crittografia asimmetrica, inventato nel 1977 da Ronald Rivest, Adi Shamir e Leonard Adleman utilizzabile per cifrare o firmare informazioni
- Il sistema di crittografia si basa sull'esistenza di due chiavi distinte, che vengono usate per cifrare e decifrare, la chiave “privata” e la chiave “pubblica”
- Queste due chiavi vengono prodotte moltiplicando insieme due numeri primi molto grandi
- La chiave “pubblica” serve per criptare, ma non puo’ essere usata per decifrare, per decrittare serve la chiave “privata”
- Per trovare la chiave privata associata ad una chiave pubblica bisognerebbe risalire ai due numeri primi originali usati per produrle
- L’unico modo per trovarle e’ “Brute force”
- Quindi e’ praticamente impossibile decifrare perche’ si dovrebbe trovare i due numeri primi originali. Per fare cio’ si puo’ solo procedere provando tutti le possibili combinazioni
- Usiamo una chiave per ogni sessione sul computer


How It Works
HTTPS


online shopping


Client
Client
Client
Server


YOUR COMPUTER
WEB SERVER
YOUR BROWSER INITIATES THE CONNECTION
I WANT AN HTTPS CONNECTION
PUBLIC KEY
PRIVATE KEY
OKAY, HERE'S MY PUBLIC KEY
A SESSION KEY IS GENERATED BY YOUR BROWSER
SESSION KEY
THE SESSION KEY IS ENCRYPTED WITH THE PUBLIC KEY
11001 00101 10011
THE ENCRYPTED SESSION KEY IS SENT TO THE SERVER
THE SERVER DECRYPTS THE SESSION KEY WITH THE PRIVATE KEY


Создание сети без доступа в интернет
Компьютер 1
Компьютер 2
Настройки сетевой карты
IP Адрес: 192.168.0.1
Маска подсети: 255.255.255.0
Шлюз: пусто
IP Адрес: 192.168.0.2
Маска подсети: 255.255.255.0
Шлюз: пусто

Decifrare RSA con computer convenzionali

- Computer, anche i piu' moderni, fanno un calcolo alla volta, "brute force"

- Il sistema di crittografia si basa sull'esistenza di due chiavi distinte, che vengono usate per cifrare e decifrare, la chiave "privata" e la chiave "pubblica".
- Queste due chiavi vengono prodotte moltiplicando insieme due numeri primi molto grandi

- Quindi e' praticamente impossibile decifrare perche' si dovrebbe trovare i due numeri primi originali. Per fare cio' si puo' solo procedere provando tutti le possibili combinazioni. E' un problema NP-completo

- Per decifrare una singolo chiave RSA-2048 ci vorrebbero, con un computer convenzionale ci vorrebbero circa 300.000 miliardi di anni (l'universo ha ~13 miliardi di anni)

- Possiamo stare abbastanza tranquilli

Decifrare RSA, o risolvere problemi NP-Hard, si puo’ pero’ con computer non convenzionali

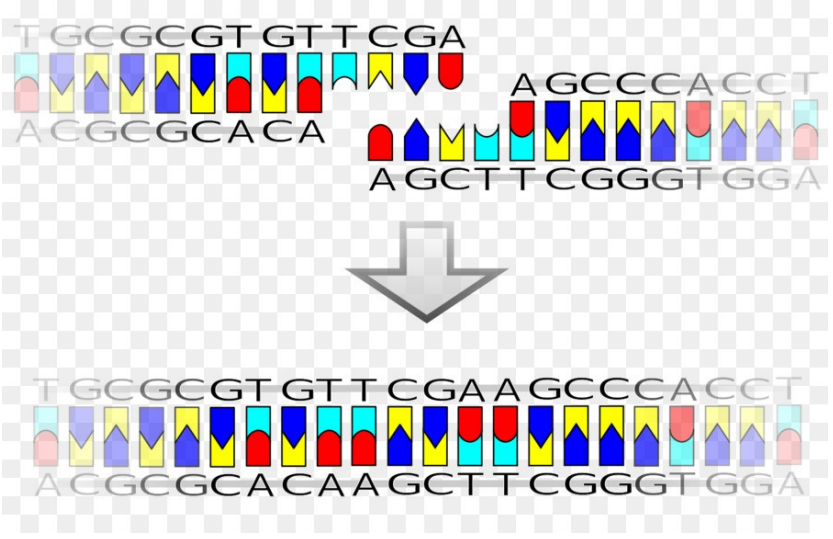
- Per risolvere problemi NP-completi ci vorrebbero calcolatori che possono fare miliardi di operazioni allo stesso momento
- Sono stati teorizzati vari tipi di computer che potrebbero fare cio’ tra cui: computer quantistici e computer a DNA (o computer chimici)
- Leonard Adleman ha dimostrato l’utilizzo del computer a DNA per risolvere problemi NP-completi nel 1992, risolvendo un’altro problema NP-completo (non il decifrare una chiave crittografica), il “Problema del commesso viaggiatore” o “problema dei sette ponti di Konigsberg” (Eulero)

Tecniche di Biologia Molecolare necessarie per svolgere calcolo

- Ligazione (reazione chimica)
- Amplificazione (PCR)
- Elettroforesi (rilevamento risultato)

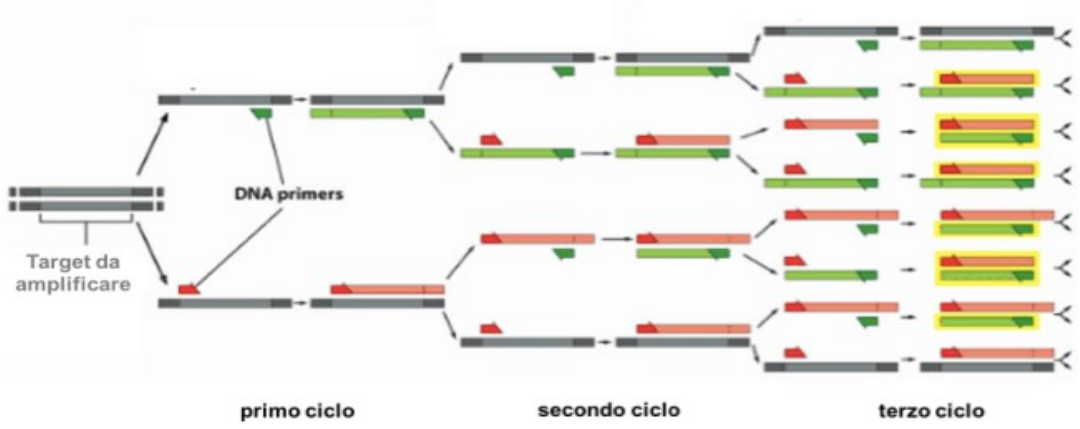
Ligazione

- Permette di “legare” insieme due pezzi di DNA che si siano appaiati



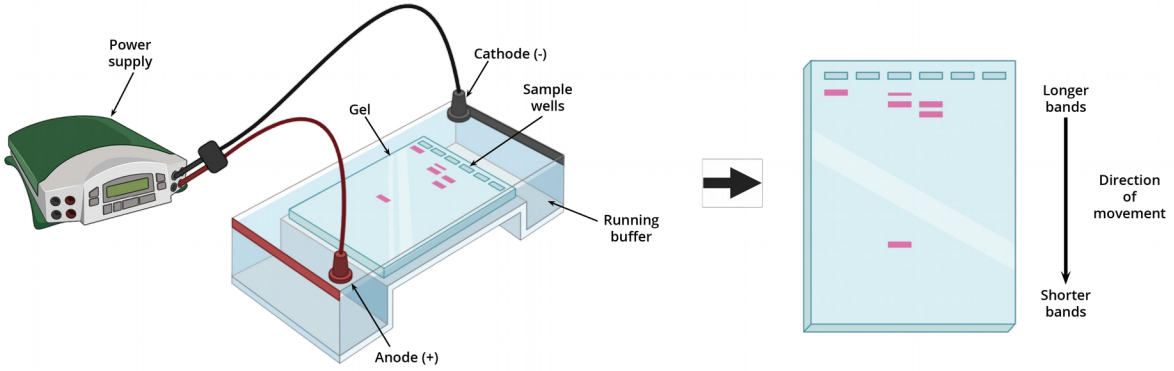
PCR (amplificazione)

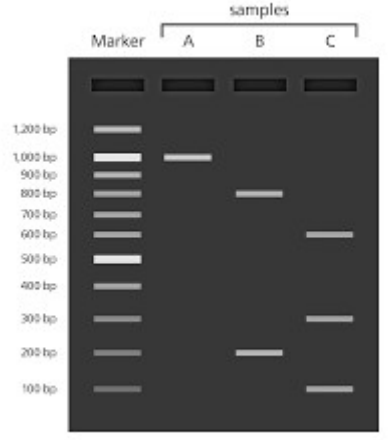
- Permette di “amplificare” DNA usando piccoli pezzetti di DNA
- Funziona a cicli, di durata variabile, ma solitamente di un paio di minuti
- Da una singola molecola di partenza dopo 30 cicli (~60 minuti) ottengo 2^{29} (536,870,912) copie



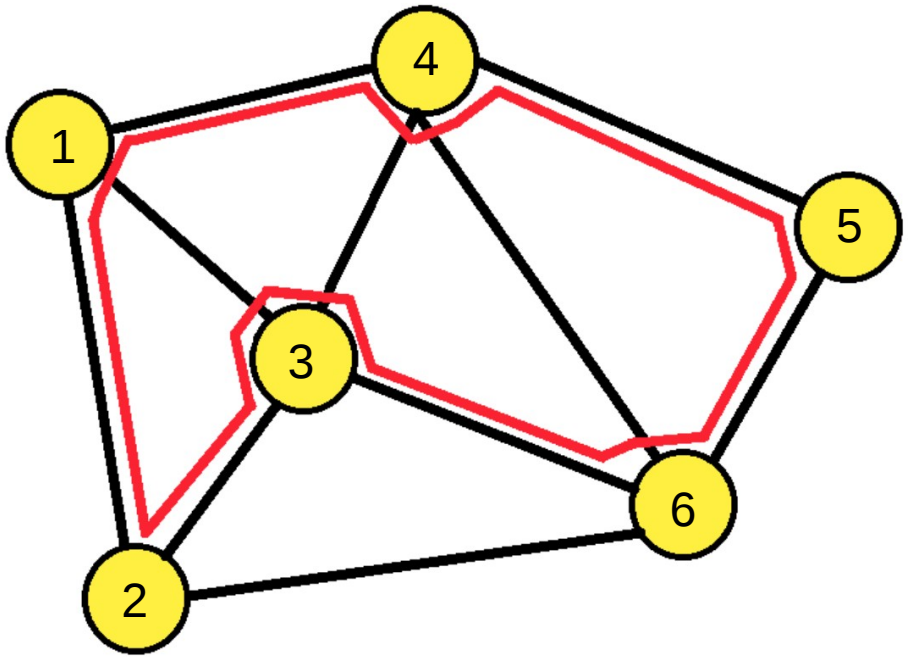
Elettroforesi

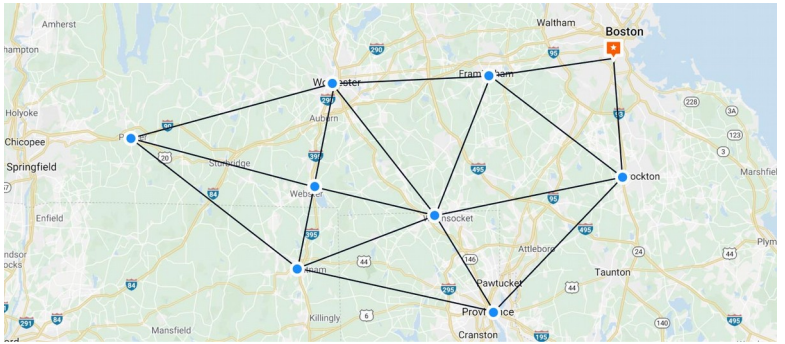
- DNA e’ carico negativamente e quindi in un campo elettrico migra verso l’anodo
- Il gel funge da settaccio, le molecole piu’ corte passano meglio attaverso le maglie
- Si puo’ cosi separare e visualizzare molecole di DNA



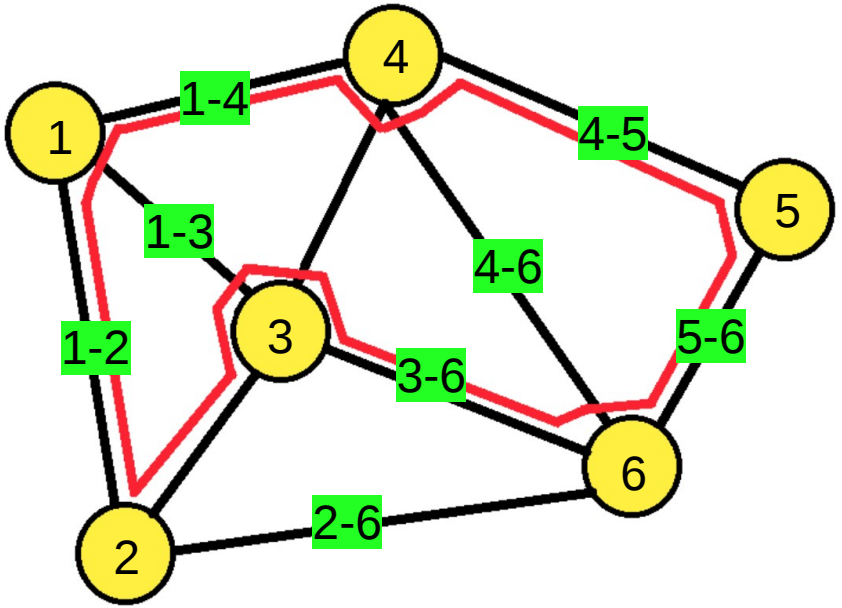


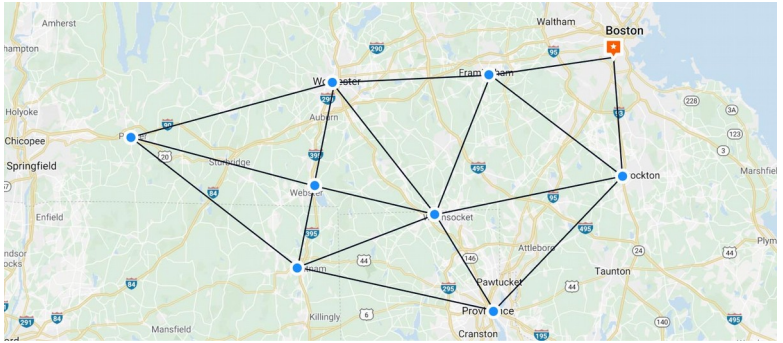
Codifica del problema



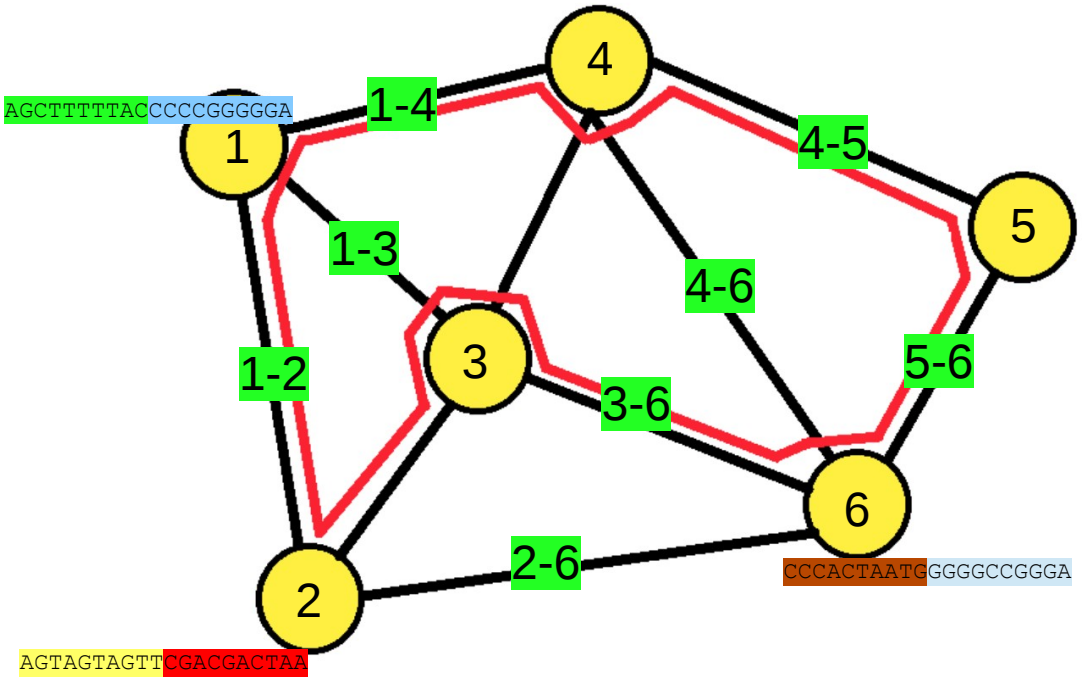


Codifica del problema



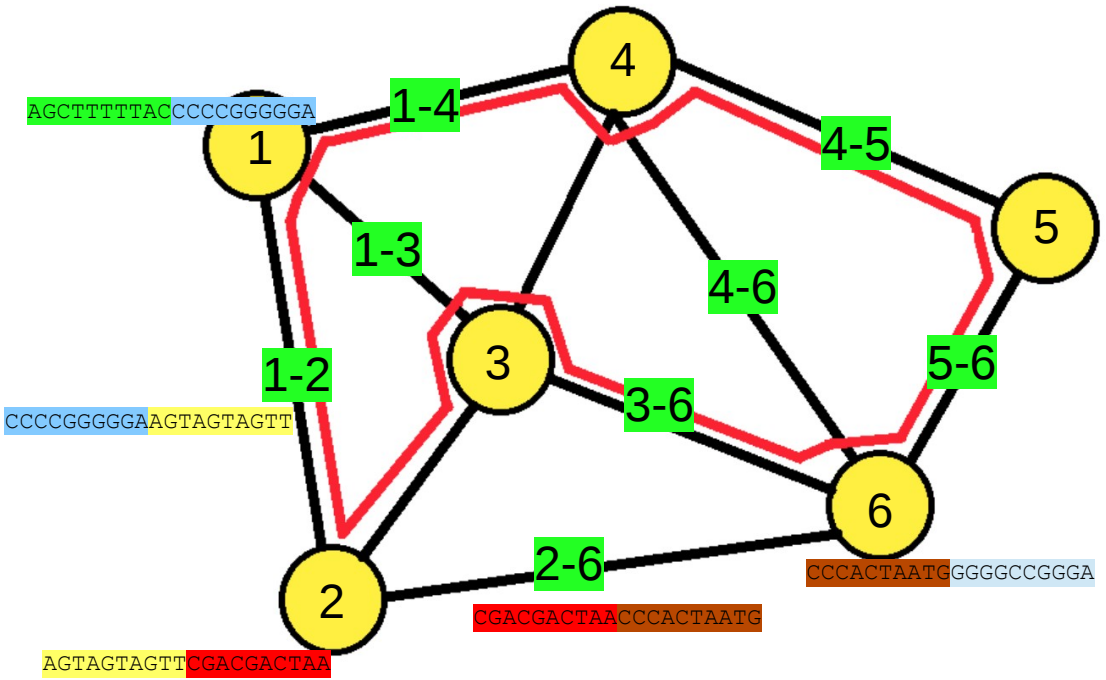


Codifica del problema



1: AGCTTTTTACCCCCGGGGGA
2: AGTAGTAGTTCGACGACTAA
3: TTCCCAGAGATACCAATTCC
4: AGGGGAAATTCACACATTAC
5: GCGCACTGACATTTCCAATA
6: CCCACTAATGGGGGCCGGGA

Codifica del problema

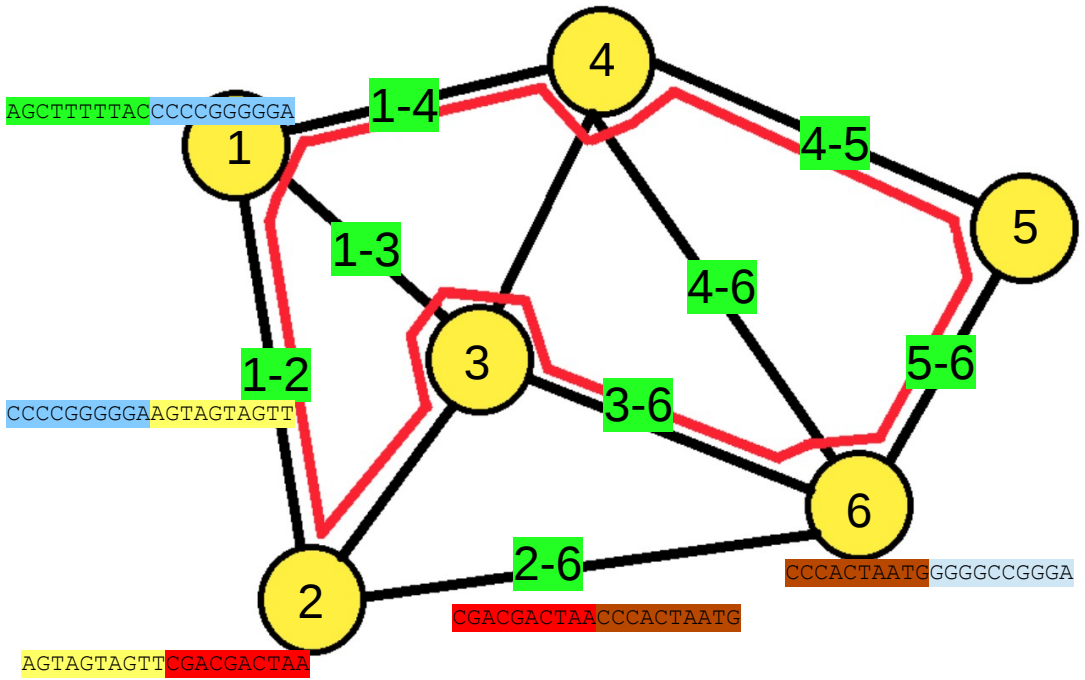


1: AGCTTTTTACCCCCGGGGGA
2: AGTAGTAGTTCGACGACTAA
3: TTCCCAGAGATACCAATTCC
4: AGGGGAAATTCACACATTAC
5: GCGCACTGACATTTCCAATA
6: CCCACTAATGGGGGCCGGGA

1-2: CCCCGGGGGAAGTAGTAGTT
2-6: CGACGACTAACCCACTAATG
….

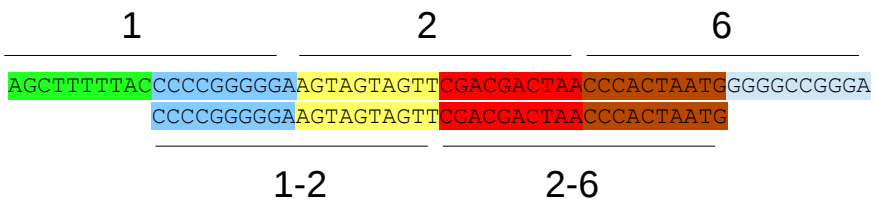
AGCTTTTTACCCCCGGGGGAAGTAGTAGTTCGACGACTAACCCACTAATGGGGGCCGGGA
CCCCGGGGGAAGTAGTAGTTCGACGACTAACCCACTAATG

Codifica del problema




```
1:  AGCTTTTTACCCCCGGGGGA
2:  AGTAGTAGTTCGACGACTAA
3:  TTCCCAGAGATACCAATTCC
4:  AGGGGAAATTCACACATTAC
5:  GCGCACTGACATTTCCAATA
6:  CCCACTAATGGGGGCCGGGA
```

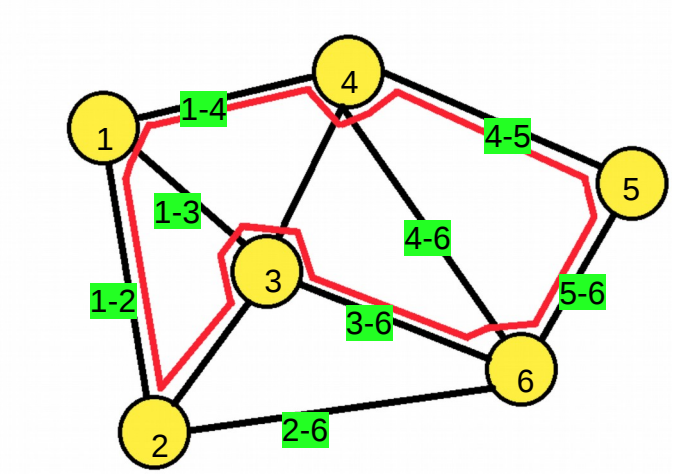
```
1-2:  CCCCGGGGGAAGTAGTAGTT
2-6:  CGACGACTAACCCACTAATG
….
```



La soluzione del problema

```
 1: AGCTTTTTACCCCCGGGGGA
 2: AGTAGTAGTTCGACGACTAA
 3: TTCCCAGAGATACCAATTCC
 4: AGGGGAAATTCACACATTAC
 5: GCGCACTGACATTTCCAATA
 6: CCCACTAATGGGGGCCGGGA

1-2: CCCCGGGGGAAGTAGTAGTT
2-6: CGACGACTAACCCACTAATG
….
```



140 di lunghezza

1 2 6 5 4 3

AGCTTTTTACCCCCGGGGGAAGTAGTAGTTCGACGACTAACCCACTAATGGGGGCCGGGAGCGCACTGACATTTCCAATA AGGGGAAATTCACACATTAC TTCCCAGAGATACCAATTCC
CCCCGGGGGAAGTAGTAGTTCGACGACTAACCCACTAATG

1-2 2-6

140 di lunghezza

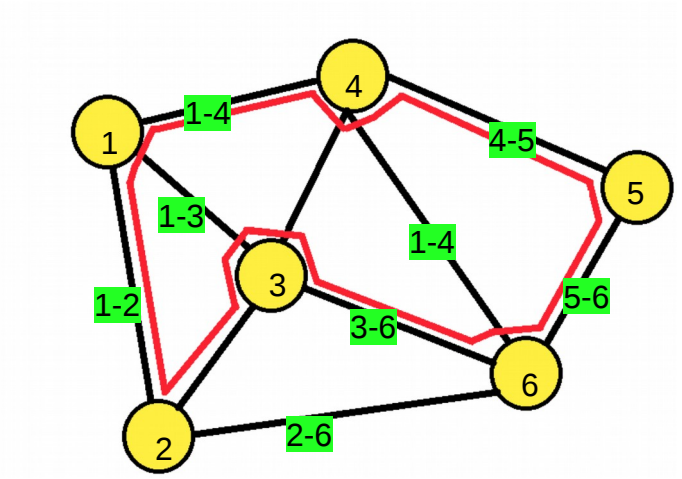
1	2	6	5	4	3

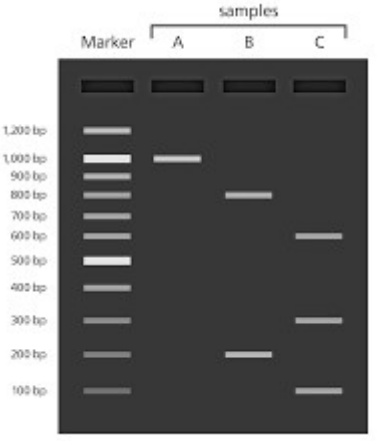
AGCTTTTTACCCCCGGGGGAAGTAGTAGTTCGACGACTAACCCACTAATGGGGGCCGGGAGCGCACTGACATTTCCAATA AGGGGAAATTCACACATTAC TTCCCAGAGATACCAATTCC

Domanda:
“Esiste almeno un percorso che parte da 1 e vi ritorno toccando ciascun punto una sola volta?”

Altre possibili soluzioni corrette:
1-2-3-6-5-4-1
1-3-2-6-5-4-1
1-4-5-6-3-2-1
……..

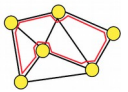
Soluzioni non corrette:
1-3-4-1-2-3-1-2-6-5-1
1-3-2-1
1-3-6-5-4-3-2-1





Workflow Sperimentale

Codifica del problema



Tempo: variabile

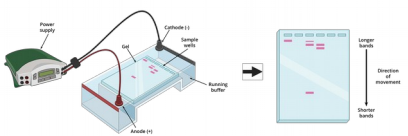
Reazione chimica che saggia
Tutte le possibili soluzioni
Contemporaneamente (ligazione)

Tempo: 1-2 ore

Amplificazione dei risultati (PCR)

Tempo: 1-2 ore

lettura risultati (elettroforesi)



Tempo: 45 min

SCIENCE • VOL. 266 • 11 NOVEMBER 1994

Molecular Computation of Solutions to Combinatorial Problems

Leonard M. Adleman

The tools of molecular biology were used to solve an instance of the directed Hamiltonian path problem. A small graph was encoded in molecules of DNA, and the "operations" of the computation were performed with standard protocols and enzymes. This experiment demonstrates the feasibility of carrying out computations at the molecular level.

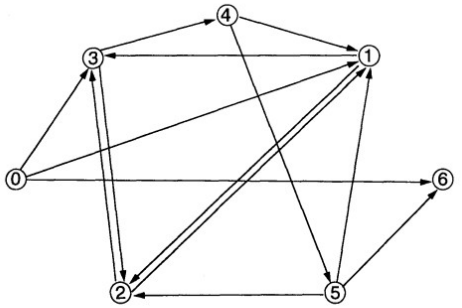
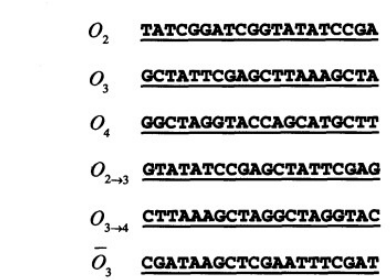


Fig. 1. Directed graph. When $v_{in} = 0$ and $v_{out} = 6$, a unique Hamiltonian path exists: 0→1, 1→2, 2→3, 3→4, 4→5, 5→6.



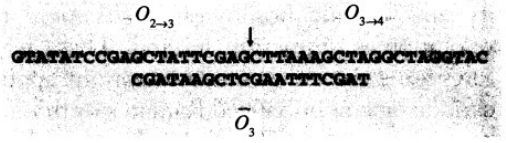


Fig. 2. Encoding a graph in DNA. For each vertex i in the graph, a random 20-mer oligonucleotide O_i is generated (shown are O_2, O_3, and O_4, for vertices 2, 3, and 4, respectively). For edge $i \to j$ in the graph, an oligonucleotide $O_{i\to j}$ is derived from the 3′ 10-mer of O_i and from the 5′ 10-mer of O_j (shown are $O_{2\to3}$ for edge 2→3 and $O_{3\to4}$ for edge 3→4). For each vertex i in the graph, $\overline{O}_i$ is the Watson-Crick complement of O_i (shown is $\overline{O}_3$, the complement of O_3). $\overline{O}_3$ serves as a splint to bind $O_{2\to3}$ and $O_{3\to4}$ in preparation for ligation. All oligonucleotides are written 5′ to 3′, except $\overline{O}_3$.

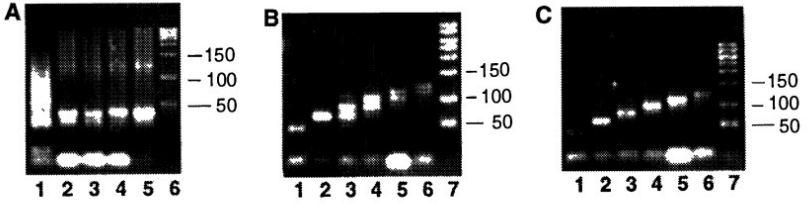
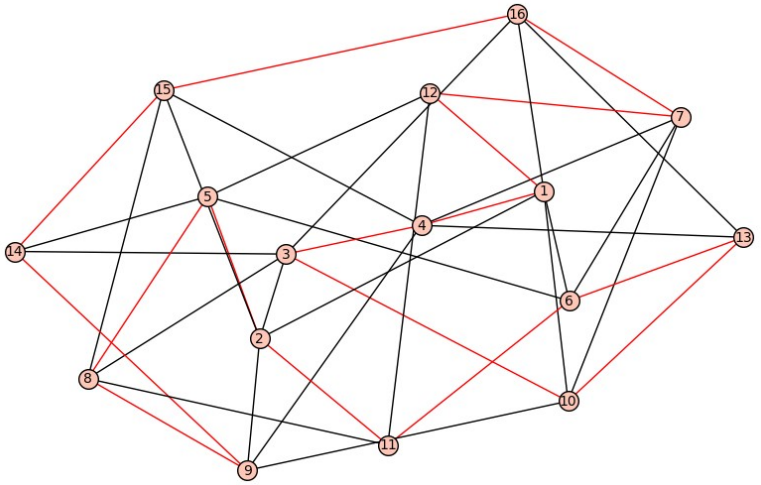
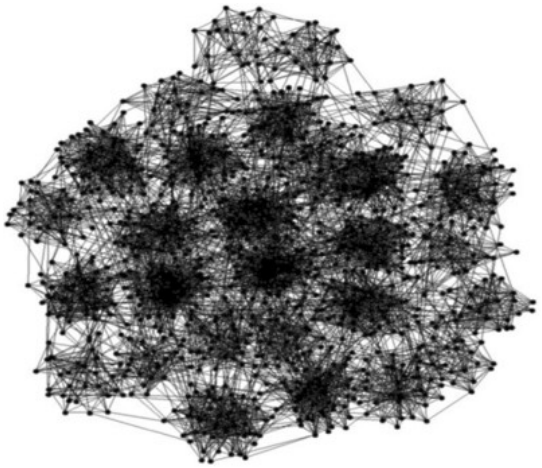
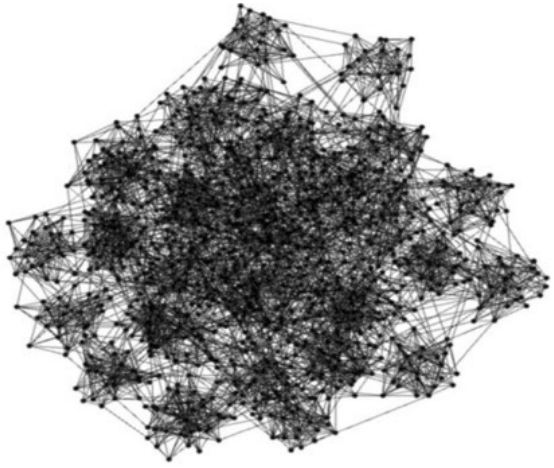


Fig. 3. Agarose gel electrophoresis of various products of the experiment. (**A**) Product of the ligation reaction (lane 1), PCR amplification of the product of the ligation reaction (lanes 2 through 5), and molecular weight marker in base pairs (lane 6). (**B**) Graduated PCR of the product from Step 3 (lanes 1 through 6); the molecular weight marker is in lane 7. (**C**) Graduated PCR of the final product of the experiment, revealing the Hamiltonian path (lanes 1 through 6); the molecular weight marker is in lane 7.

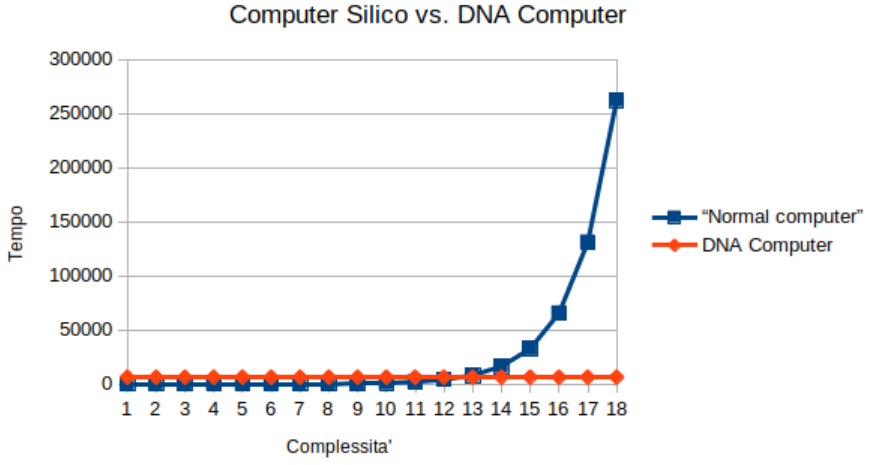




(a)



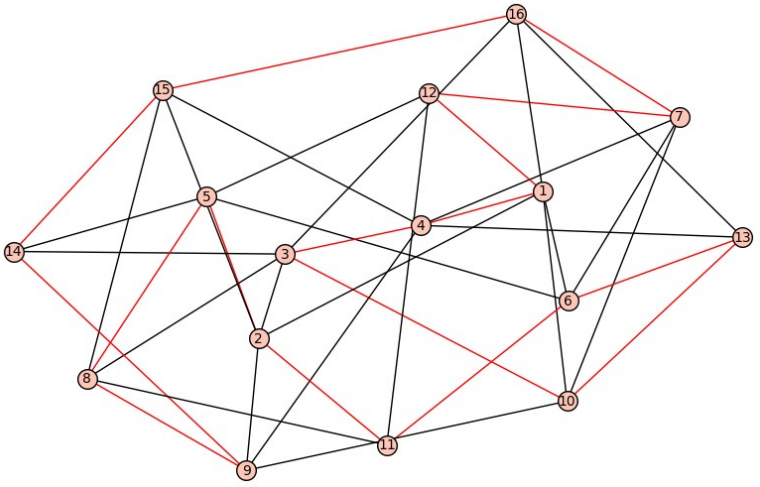
(b)



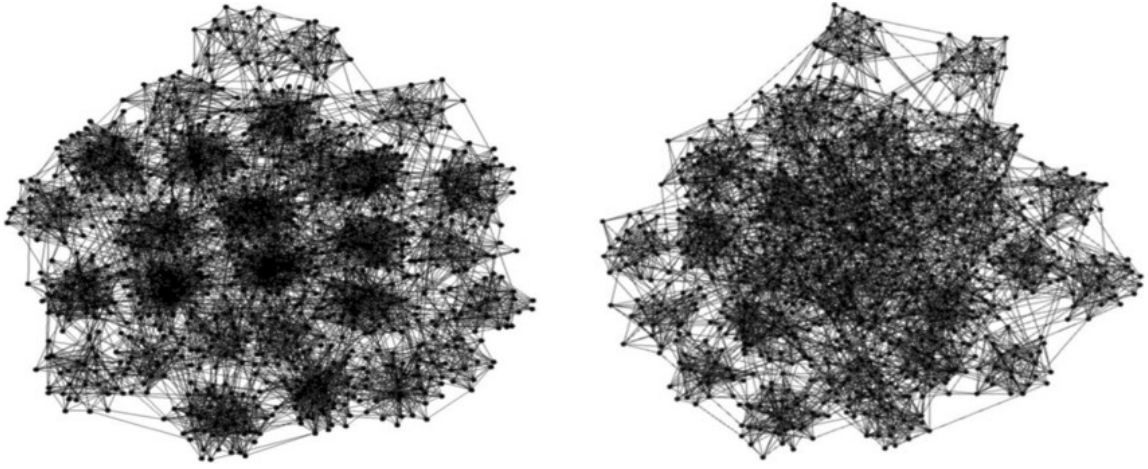
Problemi diventano velocemente “intrattabili” con computer normali perchè il tempo di esecuzione “brute force” aumenta esponenzialmente.

Il teempo diventa estremamente grande all’aumentare della complessita’.

Con calcolatori a DNA (o quantici) il tempo di esecuzione resta costante, non aumenta all’aumentare della complessita’

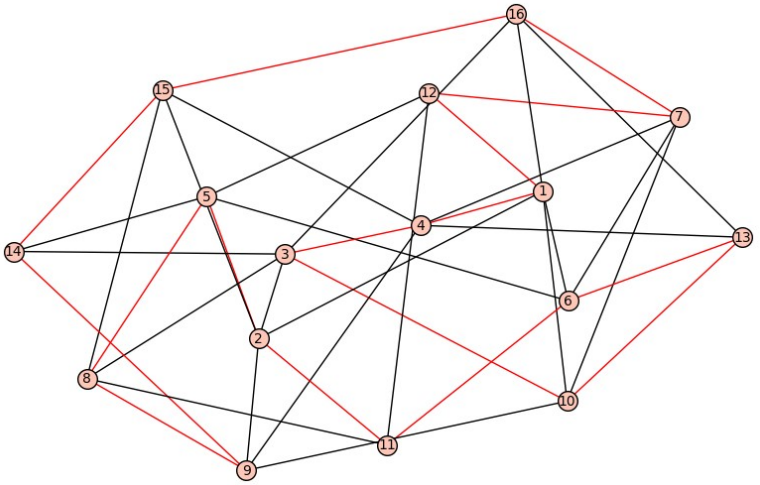


Allora perché non implementare calcolatori a DNA ,per tutti i nostri problemi, anche non NP completi?



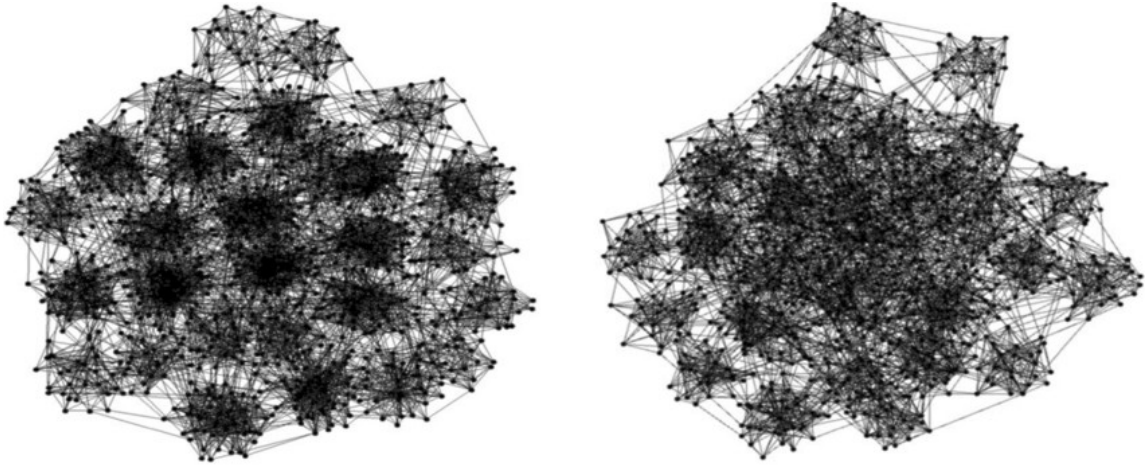
(a)

(b)

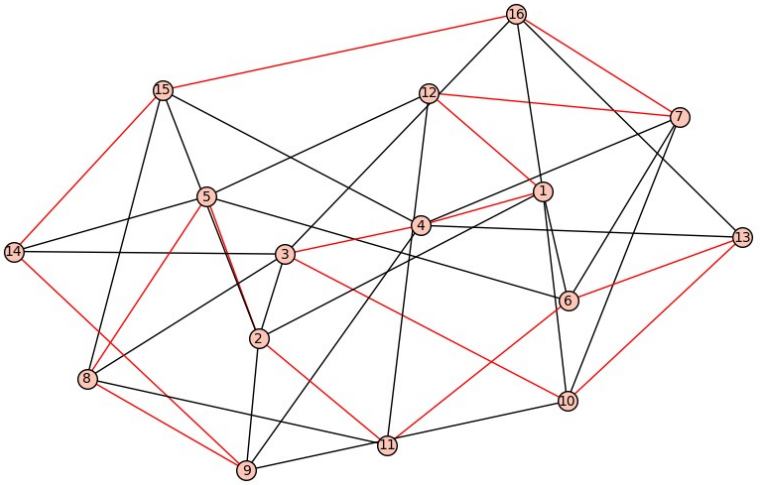


Allora perché non implementare calcolatori a DNA ,per tutti i nostri problemi, anche non NP completi?

Perchè non abbiamo il problema di tempo, ma abbiamo il problema di MASSA



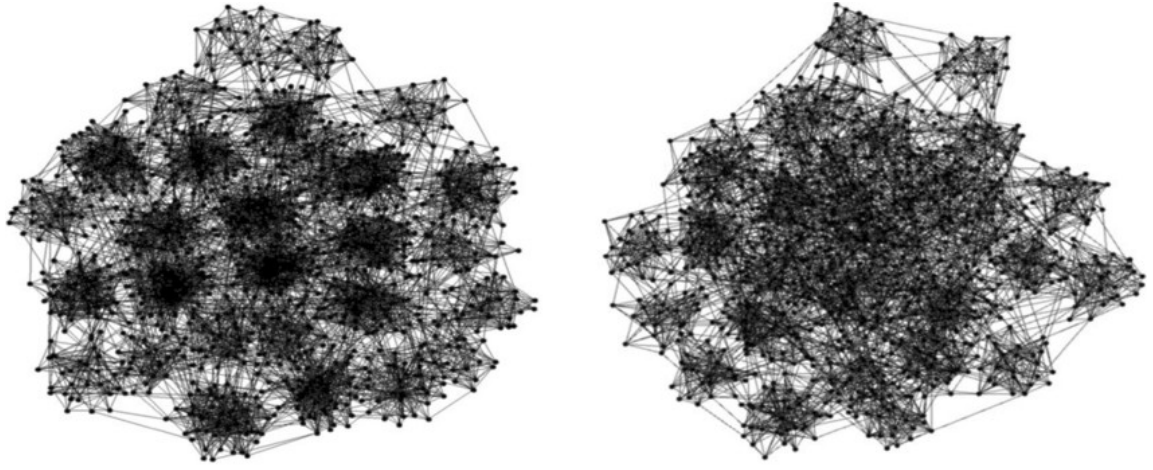
(a) (b)



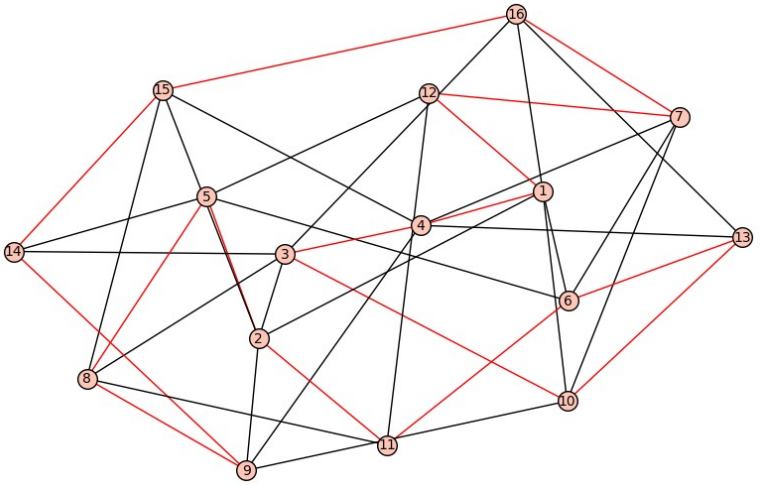
Allora perché non implementare calcolatori a DNA ,per tutti i nostri problemi, anche non NP completi?

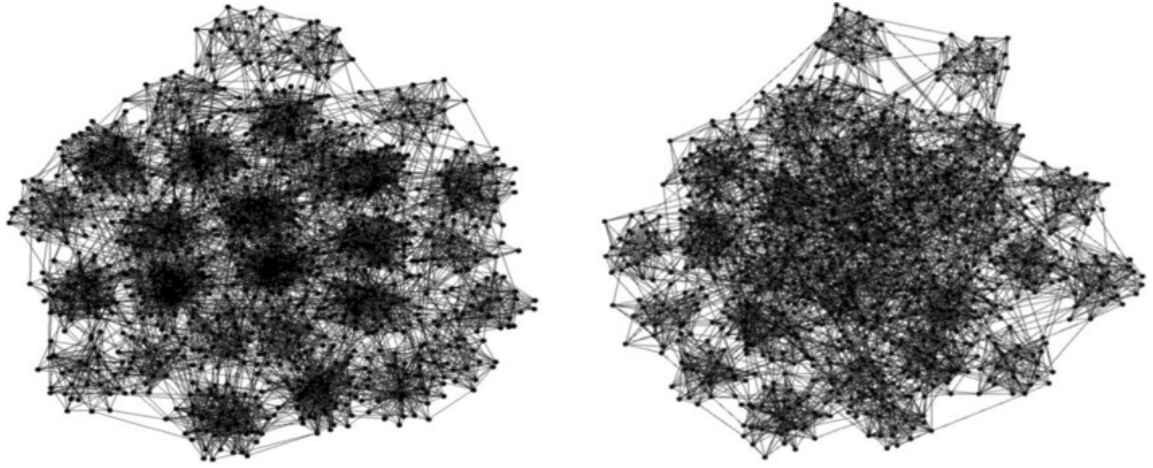
Perchè non abbiamo il problema di tempo, ma abbiamo il problema di MASSA

Problema che non sussiste con computer quantici!



(a) (b)





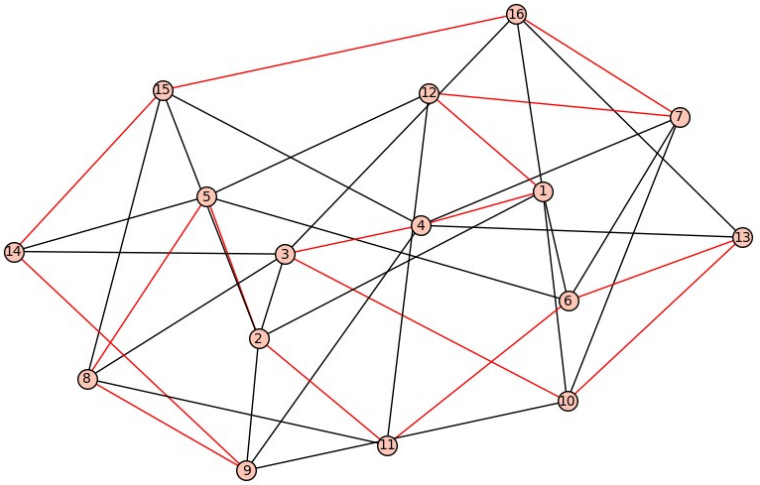
(a) (b)

Allora perché non implementare calcolatori a DNA ,per tutti i nostri problemi, anche non NP completi?

Perchè non abbiamo il problema di tempo, ma abbiamo il problema di MASSA

Problema che non sussiste con computer quantici!

Che e' un buon argomento per un futuro incontro!

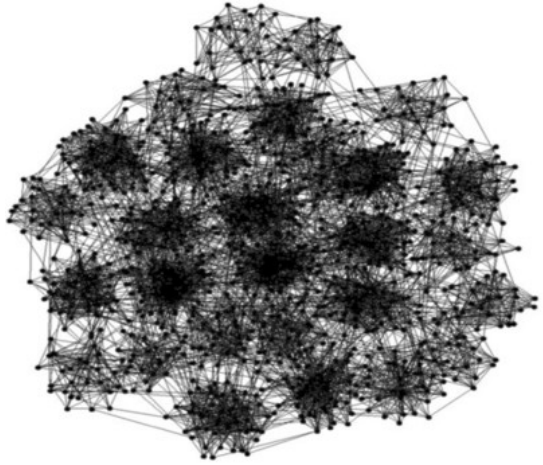


Allora perché non implementare calcolatori a DNA ,per tutti i nostri problemi, anche non NP completi?

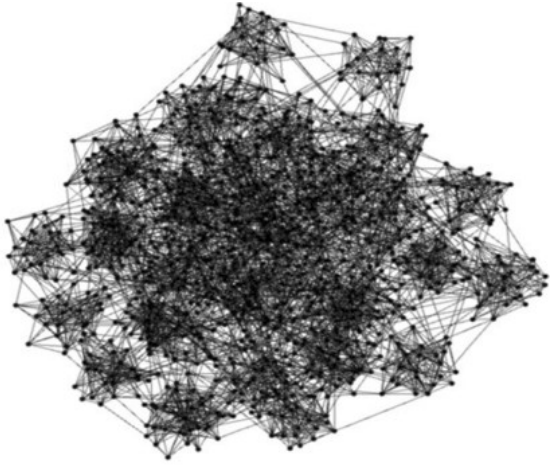
Perchè non abbiamo il problema di tempo, ma abbiamo il problema di MASSA

Problema che non sussiste con computer quantici!

Che e' un buon argomento per un futuro incontro!



(a)



(b)

GRAZIE!