UNIVERSITY OF LIVERPOOL

COMP 532
Machine Learning and BioInspired Optimization

Lecture 28: DNA Computing

Dr. Shan Luo

Department of Computer Science

shan.luo@liverpool.ac.uk

Tentative Schedule Task 2

30 April, 3pm

Group 1	**Jennifer Brown, Jessica Wimble, Andreea Perry-Gardner**	**– Article 1**
Group 2	**Ben Scotland, Callum Harris, Lauren Parker, Sam Broadhurst**	**– Article 2**
Group 3	**Michael Worthington, Jack Taylor, Robert Johnson, Michael Wright**	**– Article 3**
Group 4	**Shibao Yang, Ting Feng, Shengqiong Sun**	**– Article 4**

Tentative Schedule Task 2

11 May, 2pm

Group 5	**Orestis Katsanakis, Shaun Markham, Konstantinos Vatikiotis, Theodoros Michalopoulos**	**– Article 5**
Group 6	**Mohamed Amine Ait Mansour, Cameron Hargreaves, James Wynne**	**– Article 6**
Group 7	**Ji Jia, FengRui Zhang, QingYing Zheng, Tianda Sun**	**– Article 7**
Group 8	**Matthew Oates, Jet Lee; Dominic Edwards, Gregory Madden**	**– Article 8**

Tentative Schedule Task 2

11 May, 4pm

Group 9	**Alfredo Rodriguez Caballero, Daniel Shamaeli, Pablo Romero Salinas**	**– Article 9**
Group 10	**Lun Chen, Thomas Welsch, Haoxuan Chen**	**– Article 20**
Group 11	**Yifan Guan; Yanqing He; Zhaoye Wu**	**– Article 11**
Group 12	**Lu Chen, Keyan Li, Yuxiang Wang**	**– Article 12**
Group 13	**George Mostyn-Parry, Adrian Shannon, Robert Sherman**	**– Article 14**

Reading Material

- Leonard M. Adleman, Computing with DNA, in *Scientific American,* August 1998, pp. 54—61.

- Bonus: a comic strip explaining the basics :)

Outline of the Lecture

- Why DNA computing?
- Biochemistry basics
- Adleman's Hamiltonian path problem
- Limitations

Based on slides by Deepthi Bollu, Lehigh University, USA

What is DNA Computing?

- DNA computing is utilizing the property of DNA for **massively parallel computation**
- With an appropriate setup and enough DNA, one can potentially solve huge problems by **parallel search**
- Utilizing DNA for this type of computation can be much faster than utilizing a conventional computer

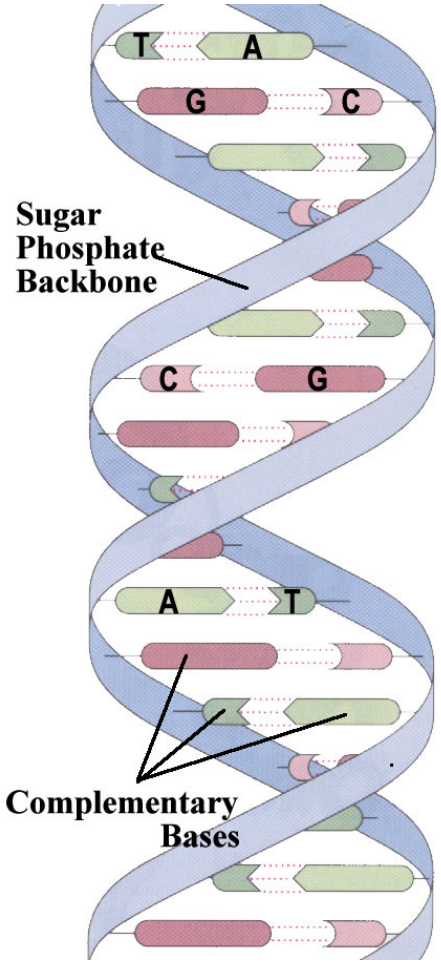
- Leonard Adleman proposed that the makeup of DNA and its multitude of possible combining nucleotides could have application in computational research techniques

What is DNA?

- **DNA** stands for **D**eoxyribo**n**ucleic **A**cid
- It represents the genetic blueprint of living creatures
- It contains “instructions” for assembling cells

- Every cell in human body has a complete set of DNA
- DNA is unique for each individual

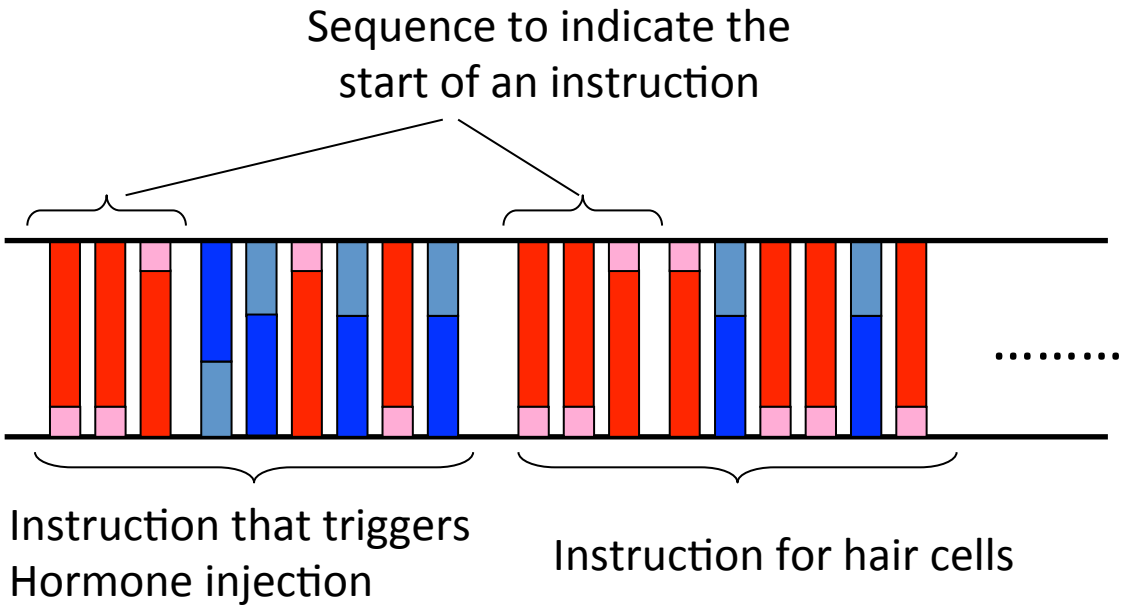
Double Helix



- "*Sides*"
  Sugar-phosphate backbones
- "*Ladders*"
  complementary base pairs
  - **A**denine & **T**hymine
  - **G**uanine & **C**ytosine
- Two strands are held together by weak hydrogen bonds between the complementary base pairs

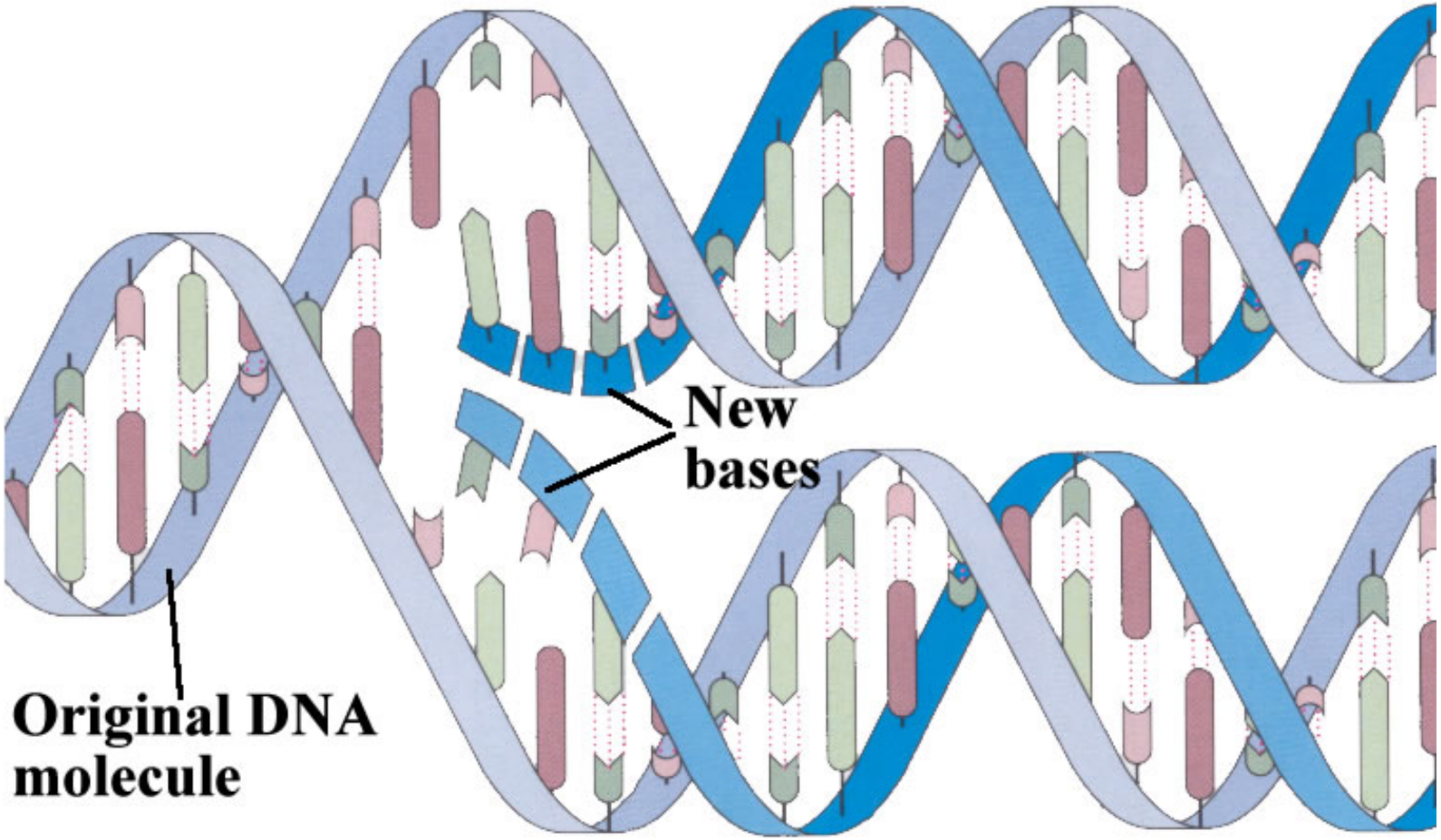
Source: "Human Physiology: From Cells to System 4th Ed.", L. Sherwood, Brooks/Cole, 2001, C-3

Instructions in DNA



- Instructions are *coded* in a sequence of the DNA bases
- A segment of DNA is exposed, transcribed, and translated to carry out instructions

DNA Duplication



Source: “Human Physiology: From Cells to System 4th Ed.”, L. Sherwood, Brooks/Cole, 2001, C-5

Why DNA Computing?

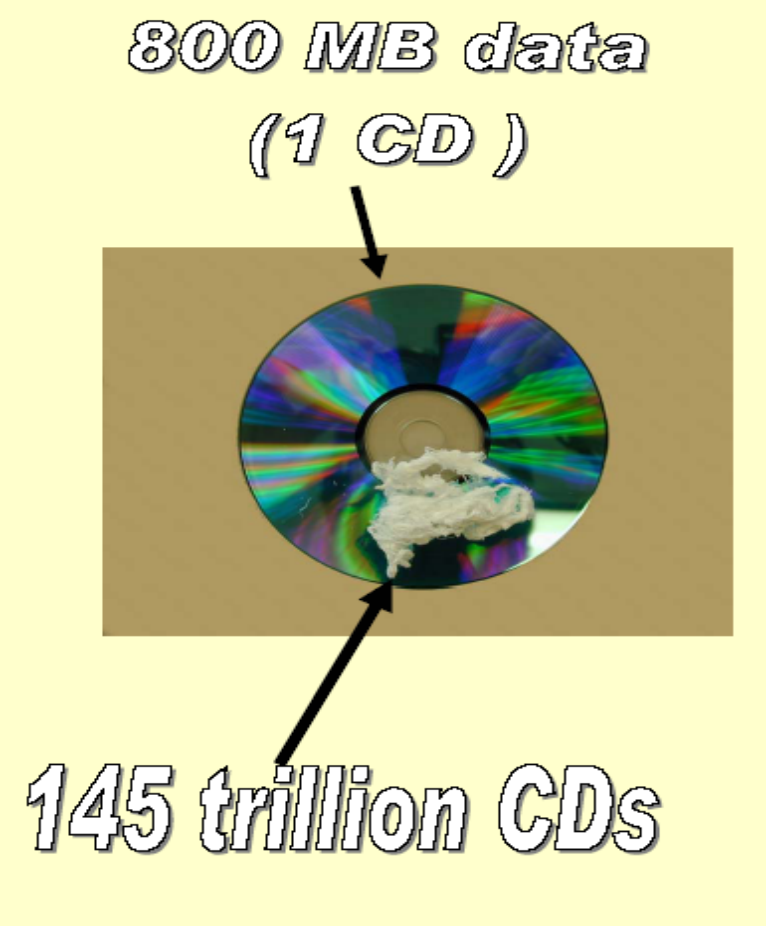
"Computation *using* DNA"
not "computation *on* DNA"

- Initiated in 1994 by Dr. Leonard M. Adleman

- Why DNA:
  - Extremely dense information storage
  - Enormous parallelism
  - Extraordinary energy efficiency

Dense Information Storage

- This image shows 1 gram of DNA on a CD. The CD can hold 800 MB of data.
- The 1 gram of DNA can hold about $1x10^{14}$ MB of data.
- The number of CDs required to hold this amount of information, lined up edge to edge, would circle the Earth 375 times, and would take 163,000 centuries to listen to.



Massively Parallel

- A test tube of DNA can contain trillions of strands
- Each operation on a test tube of DNA is carried out **on all strands in the tube in parallel**

- In Adleman's experiment, 10^{14} operations were carried out in parallel

Energy Efficiency

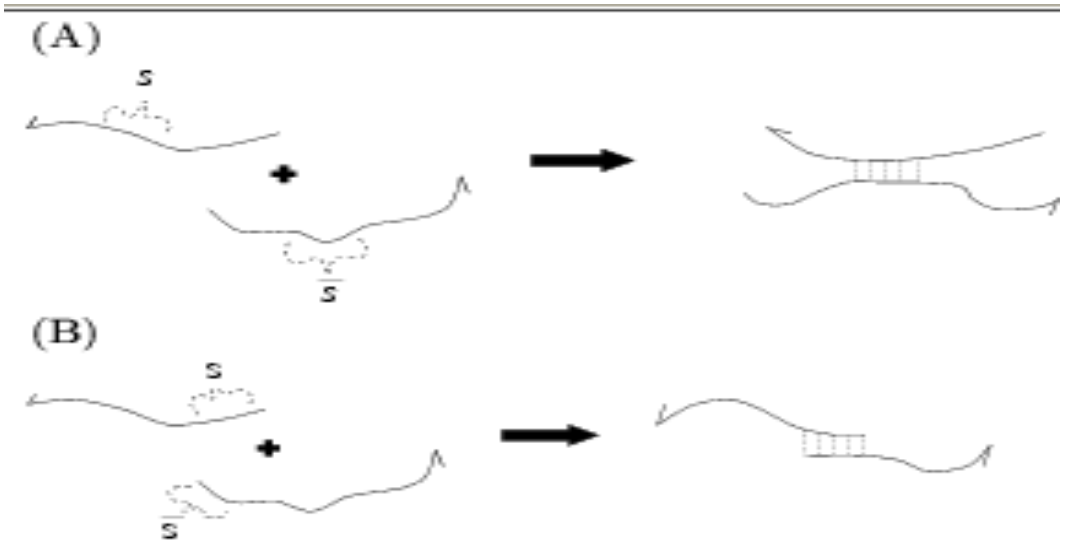
- Adleman figured his computer was running 2×10^{19} operations per joule

- In comparison: the current most efficient supercomputers reach 9×10^{9} operations per joule

Biochemistry Basics

- Basic operations needed to work with DNA:
  - **Annealing**
    (combining complementary strands)
  - **Polymerase Chain Reaction** (PCR)
    (amplifying DNA)
  - **Gel Electrophoresis**
    (measuring the length of DNA)
  - **Extraction**
    (finding strands that contain a specific sequence)

Annealing



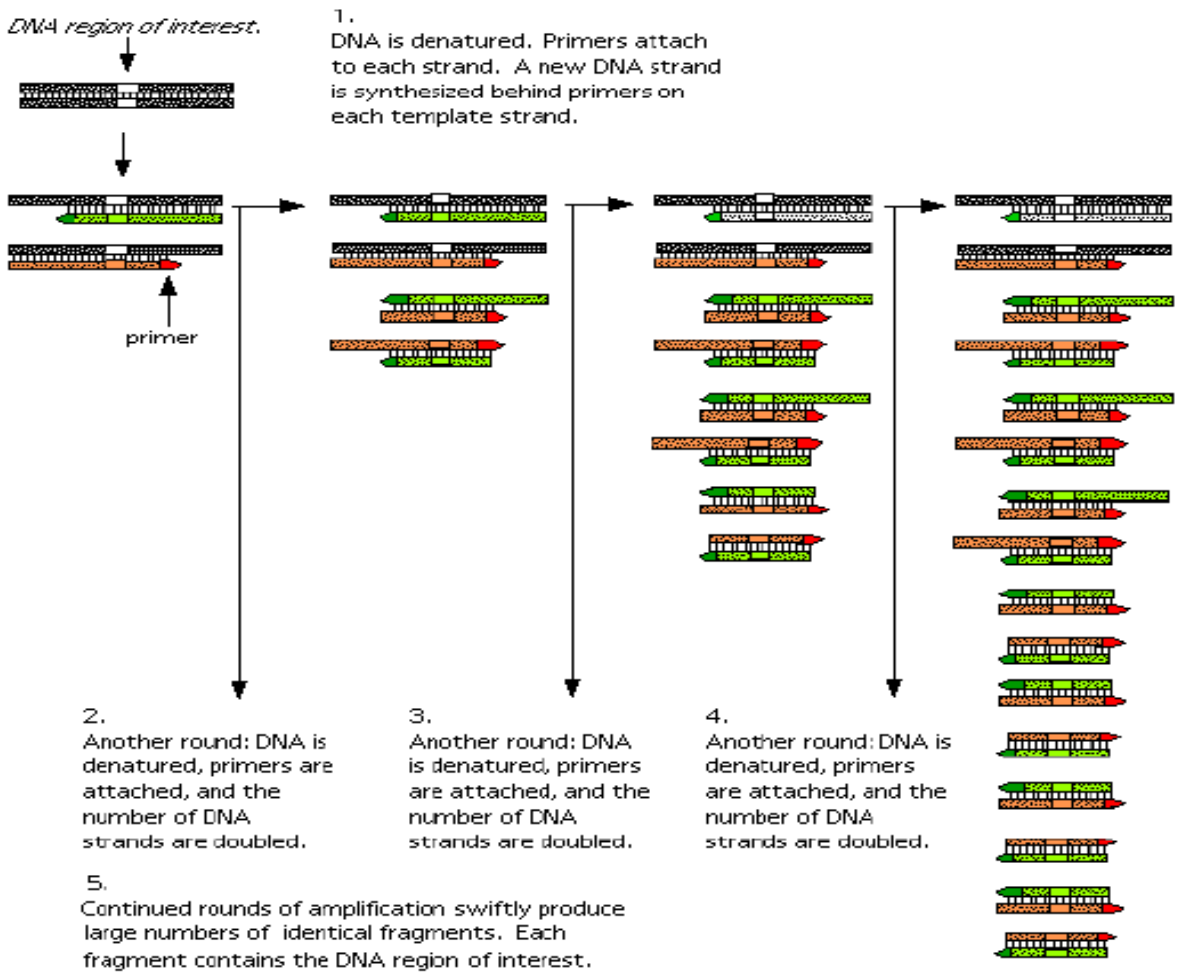
- The hydrogen bonding between two complimentary sequences is weaker than the one that links nucleotides of the same sequence.
- It is possible to pair (anneal) and separate (melt) two antiparallel and complementary single strands.

Polymerase Chain Reaction

PCR: One way to amplify DNA.

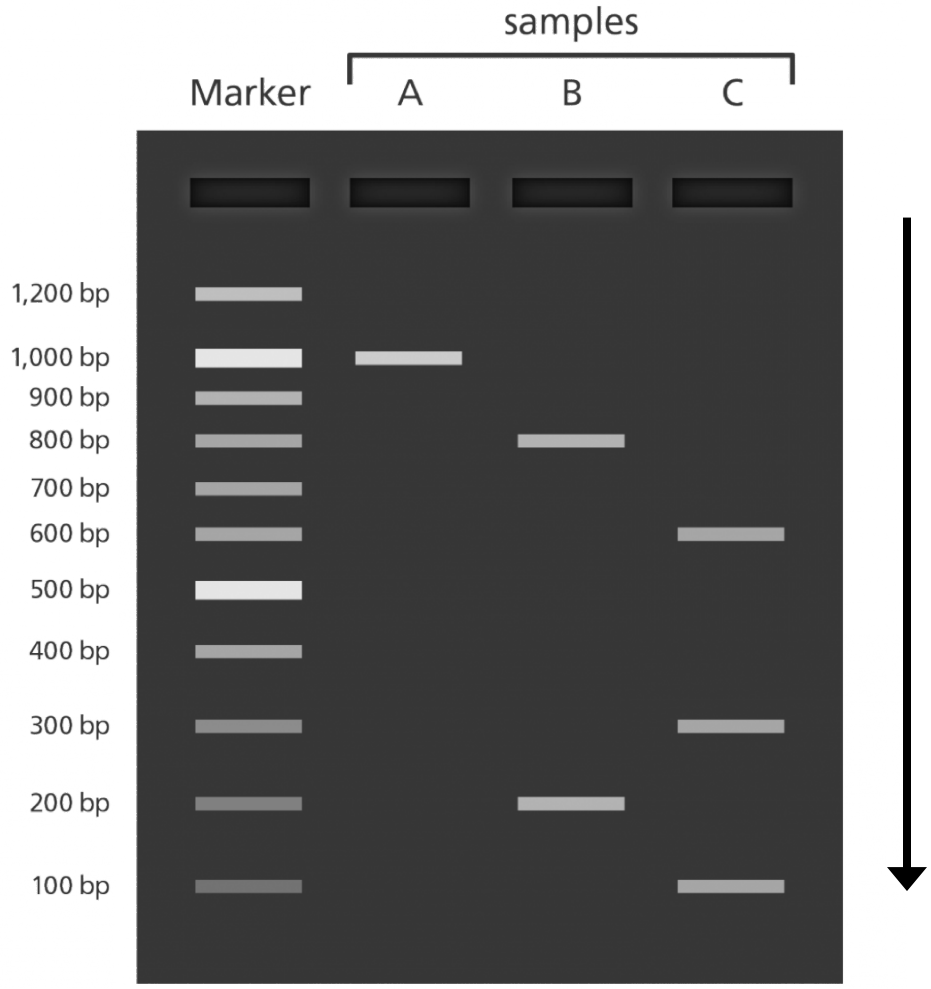
PCR alternates between two phases: separate DNA into single strands using heat; convert into double strands using *primer* and polymerase reaction.

PCR rapidly amplifies a single DNA molecule into billions of molecules



Gel Electrophoresis

- Used to measure the length of a DNA molecule.
- Based on the fact that DNA molecules are negatively charged.



Extraction

- Annealing of complimentary strands can be used for fishing out target molecules from a sample S.

1. Denature the double stranded molecules.
2. Attach probe to a filter and pour the solution S through it.
   - The probe for s molecules would be $\bar{s}$.
3. Double stranded molecules are fixed to the filter.
4. Filter is then denatured and only target molecules remain.

- Adleman attached probes to magnetic beads.

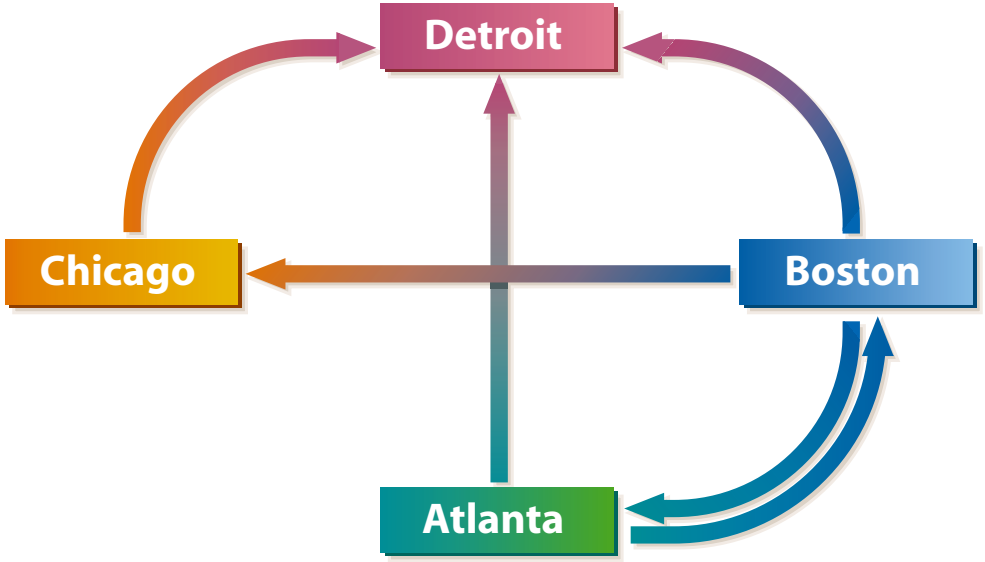
The Hamiltonian Path Problem

- A directed graph $G = (V, E)$
  - $|V|=n$, $|E|=m$
  - two distinguished vertices $V_{in} = s$ and $V_{out} = t$.

- Verify whether there is a path $(s, v1, v2, \ldots, t)$
  - which is a sequence of "one-way" edges that begins in V_{in} and ends in V_{out}
  - whose length (in no. of edges) is n-1 (i.e. enters all vertices)
  - Whose vertices are all distinct (i.e. enters every vertex exactly once.)

A CLASSIC NP-COMPLETE PROBLEM

Example



- V_{in} = Atlanta; V_{out} = Detroit
- Can you find the Hamiltonian Path?

Atlanta → Boston → Chicago → Detroit

Brute Force Algorithm

- Brute force algorithm is to
  - Generate ***all possible paths*** with exactly n-1 edges
  - Verify whether one of them obeys the problem constraints.

- Problem: How many paths can there be?

There are **(n-2)!** permutations of the (n-2) intermediate cities!

- Not practical on a normal computer
- However: DNA computing is massively parallel!

Adleman's Experiment

- Makes use of the DNA molecules to solve HPP.
- Each path can be generated *independent* of all others, allowing "*Parallelism*". But: "*non-determism*" as well.

- Number of Lab procedures grows *linearly* with the number of vertices in the graph.
  - Linear due to the fact that an exponential number of operations are done in *parallel*.

- At the heart, it is a brute force algorithm executing an exponential number of operations.

Algorithm (non-deterministic)

1. Generate random paths
2. From all paths created in step 1, keep only those that start at s and end at t.
3. From all remaining paths, keep only those that visit exactly n vertices.
4. From all remaining paths, keep only those that visit each vertex at least once.
5. If any path remains, return “yes”; otherwise, return “no”.

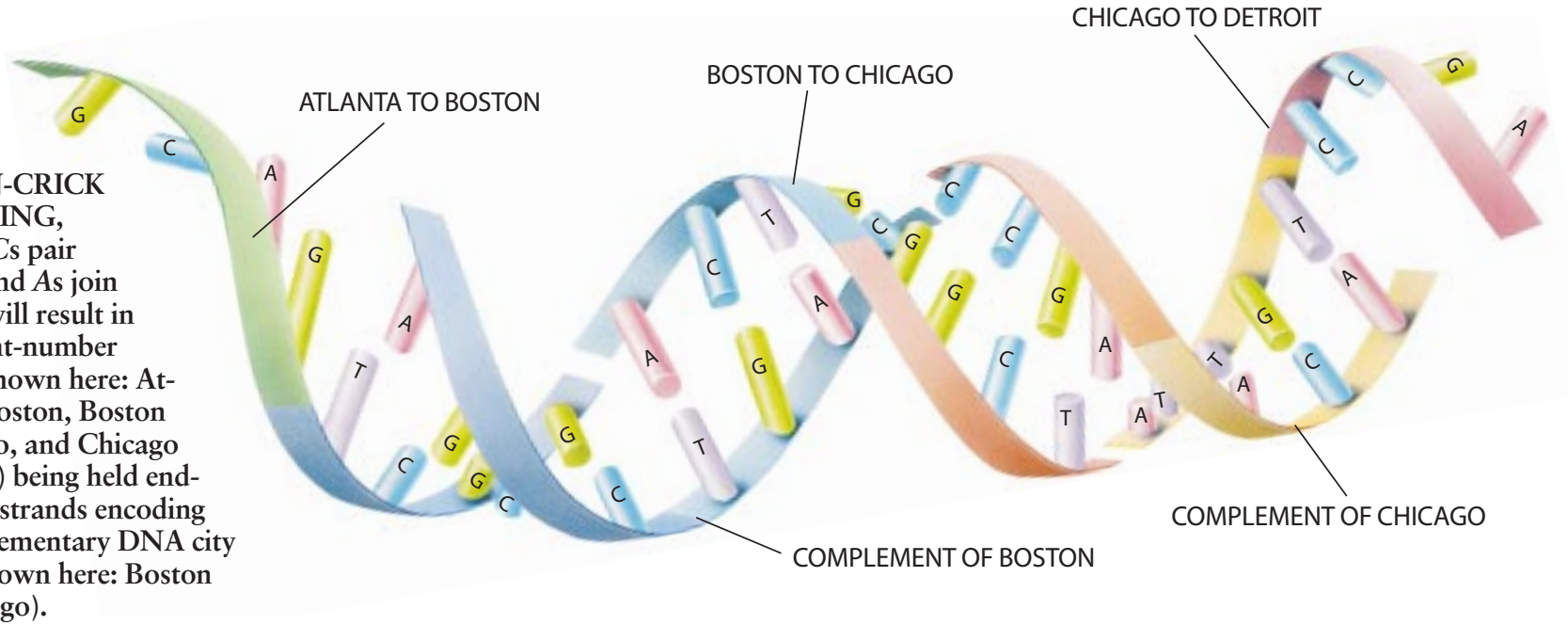
Step 1: Random Path Generation

CITY	DNA NAME	COMPLEMENT
ATLANTA	ACTTGCAG	TGAACGTC
BOSTON	TCGGACTG	AGCCTGAC
CHICAGO	GGCTATGT	CCGATACA
DETROIT	CCGAGCAA	GGCTCGTT

FLIGHT	DNA FLIGHT NUMBER
ATLANTA - BOSTON	GCAGTCGG
ATLANTA - DETROIT	GCAGCCGA
BOSTON - CHICAGO	ACTGGGCT
BOSTON - DETROIT	ACTGCCGA
BOSTON - ATLANTA	ACTGACTT
CHICAGO - DETROIT	ATGTCCGA

- **Edge**: last part of first city + first part of last city
- Edges **anneal** with complement of cities

Annealing of Path Building Blocks



WATSON-CRICK ANNEALING, in which *C*s pair with *G*s and *A*s join with *T*s, will result in DNA flight-number strands (shown here: Atlanta to Boston, Boston to Chicago, and Chicago to Detroit) being held end-to-end by strands encoding the complementary DNA city names (shown here: Boston and Chicago).

- Adleman added a "pinch" (about 10^{14}) molecules of each building block to a test tube.
- In one second, *approximately* all possible paths were generated!

Step 2: Correct Start and End Cities

- The result of step 1 is amplified by **PCR** using primers that are the **complements** of the **last part of the start city**, and the **first part of the end city**.

- This way, only those molecules encoding paths that have the correct start and end city are amplified.

Step 3: Correct Path Length

- **Gel Electrophoresis** is used to order the result of step 2 by length, and keep only those molecules that encode the correct number of cities.

- In our example:
  - each edge consists of 8 bases
  - we have n = 4 cities, so (n-1) = 3 edges
  - we are looking for paths of length 24.

Step 4: Visit Each City

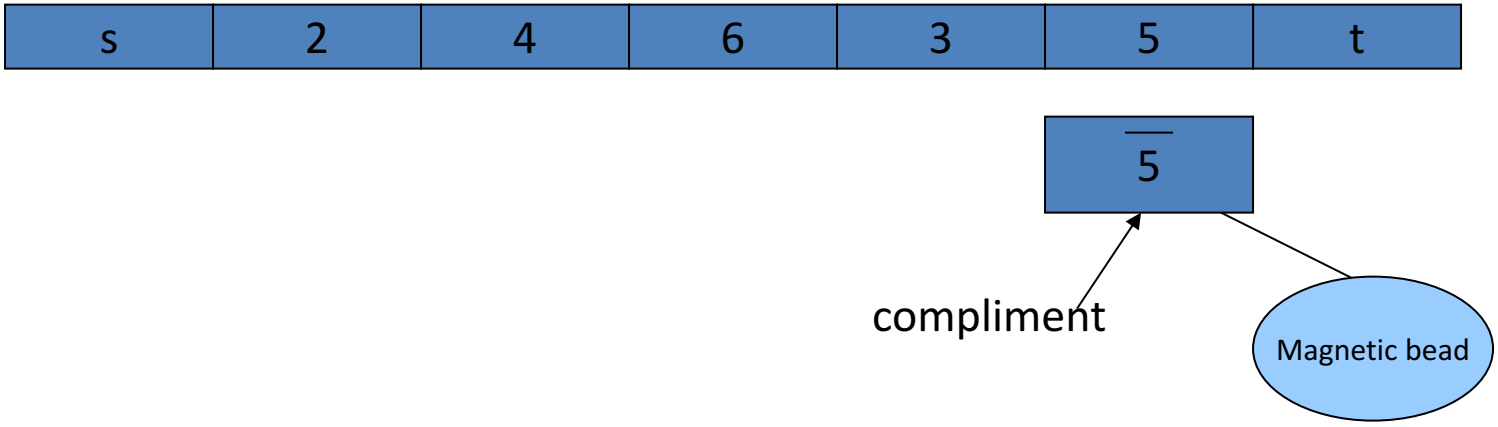
- For each intermediate city, a **probe** is used to extract paths that pass through that city
  - The probe is the complement of the city

- Doing this for each city in turn, and discarding the remainder at each step, ensures the final solution contains only paths that pass through each city

- Combined with step 3, this ensures that the remaining paths **pass through each city exactly once**!

Step 4: Visit Each City

- This technique is called **Affinity Purification**.
- Adleman used magnetic beads attached to the probe to filter the correct molecules.



Step 5: Check the Answer

- If any molecules remain after step 4, this means a path exists.
- PCR is applied again to amplify the result.
- If DNA is present, we have a solution!

- But.. what *is* the solution?

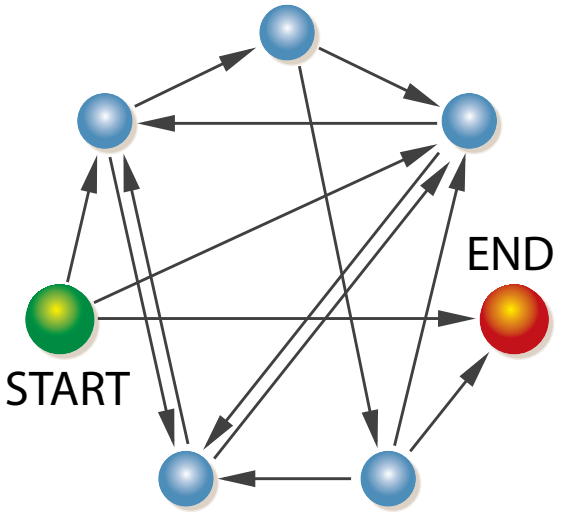
Obtaining the Answer

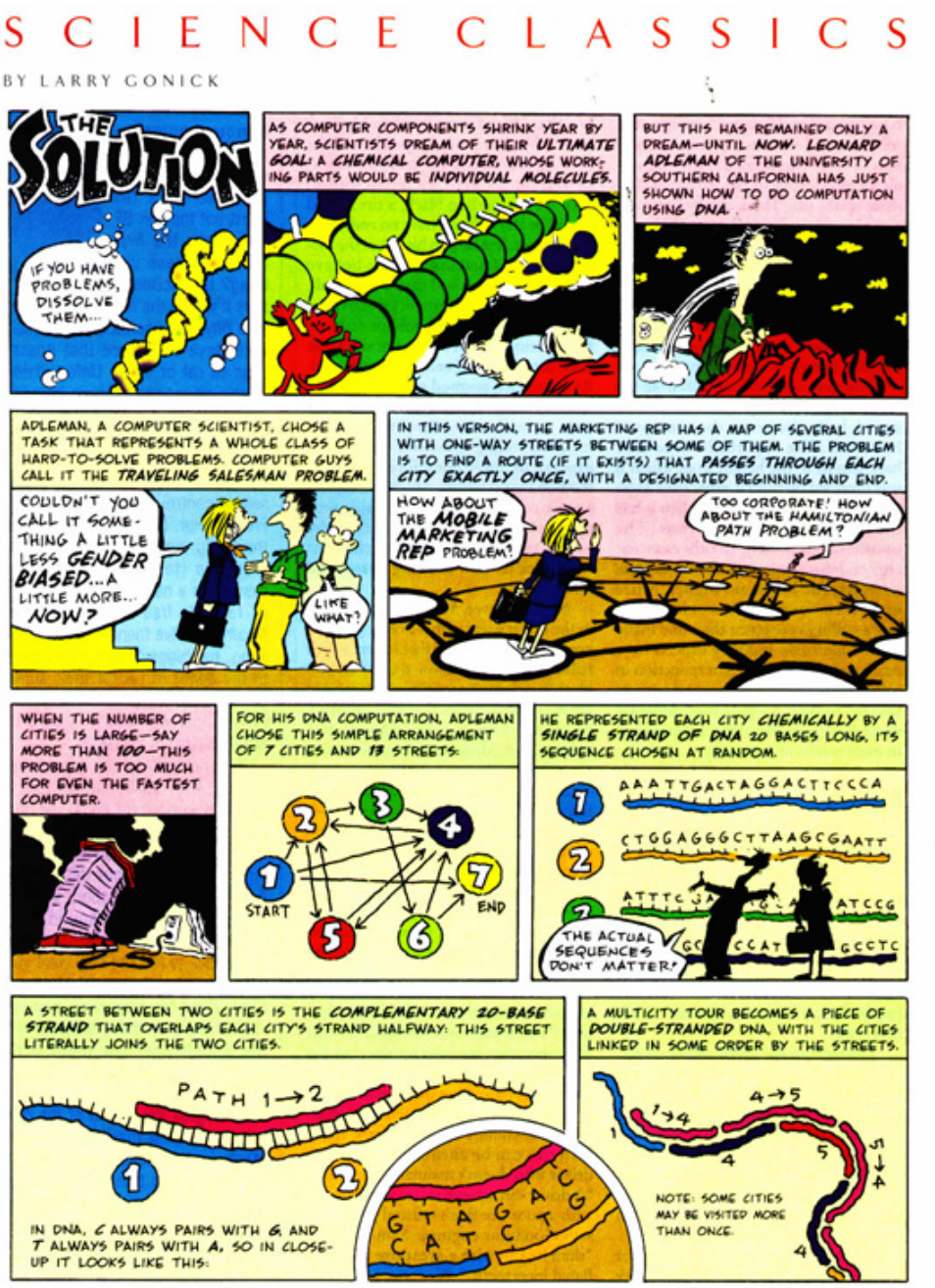
- Conduct a “graduated PCR” using a series of PCR amplifications
  - Use primers for the start and some city k.
  - Measure the length of the resulting DNA molecules using Gel Electrophoresis.
  - Divided by the length of each edge, this tells you the position of city k in the path.

- Do this for every intermediate city to obtain the answer.

Adleman's Experiment

- Adleman used a problem consisting of 7 cities and 14 edges.
- Each city was encoded by a DNA sequence of length 20.
- The solution was a path of length 120.





Discover magazine published an article in comic strip format about Leonard Adleman's discovery of DNA computation. Not only entertaining, but also a very clear explanation of molecular computation.

(see VITAL)

Recap

1. Generate random paths
   → Annealing
2. From all paths created in step 1, keep only those that start at s and end at t.
   → Polymerase Chain Reaction
3. From all remaining paths, keep only those that visit exactly n vertices.
   → Gel Electrophoresis
4. From all remaining paths, keep only those that visit each vertex at least once.
   → Affinity Purification with magnetic beads
5. If any path remains, return “yes”; otherwise, return “no”.
   → Polymerase Chain Reaction

Potential Problems

- There are 2 problems with extraction:
  - The removal of strands containing the sequence is not 100% efficient.
  - May at times inadvertently remove strands that do not contain the specified sequence.

- Adleman did not encounter problems with extraction because only a few operations were required.
- However, for a large problem instance, the number of extractions required may run into hundreds or even thousands.

Potential Problems

- What would happen if a ‘good’ path were lost during one of the extraction operations in step 4?
  - **FALSE NEGATIVE!**
  - Adleman’s solution: to amplify the content of the test tube.

- What if a ‘bad’ path is taken as if it were ‘good’?
  - **FALSE POSITIVE!**
  - Less dangerous, because the answer could be verified at the end of the computation.

Limitations

- The computation time required to solve problems with a DNA computer does not grow exponentially, but amount of DNA required DOES.
  - Tuning the experiment may take weeks.
  - Computation is fast, but obtaining the answer may be very slow.

- DNA computing involves a relatively large chance of error.
  - As size of problem grows, probability of receiving incorrect answer eventually becomes greater than probability of receiving correct answer

The Future

- Adleman's algorithm for the HPP was simple. As technology becomes more refined, more efficient algorithms may be discovered.

- DNA Manipulation technology has rapidly improved in recent years, and future advances may make DNA computers more efficient.

- DNA computers are *unlikely* to feature word processing, emailing and solitaire programs.

- Instead, their powerful computing power will be used for areas of encryption, genetic programming, language systems, and algorithms or by airlines wanting to map more efficient routes. Hence better applicable in only some promising areas.