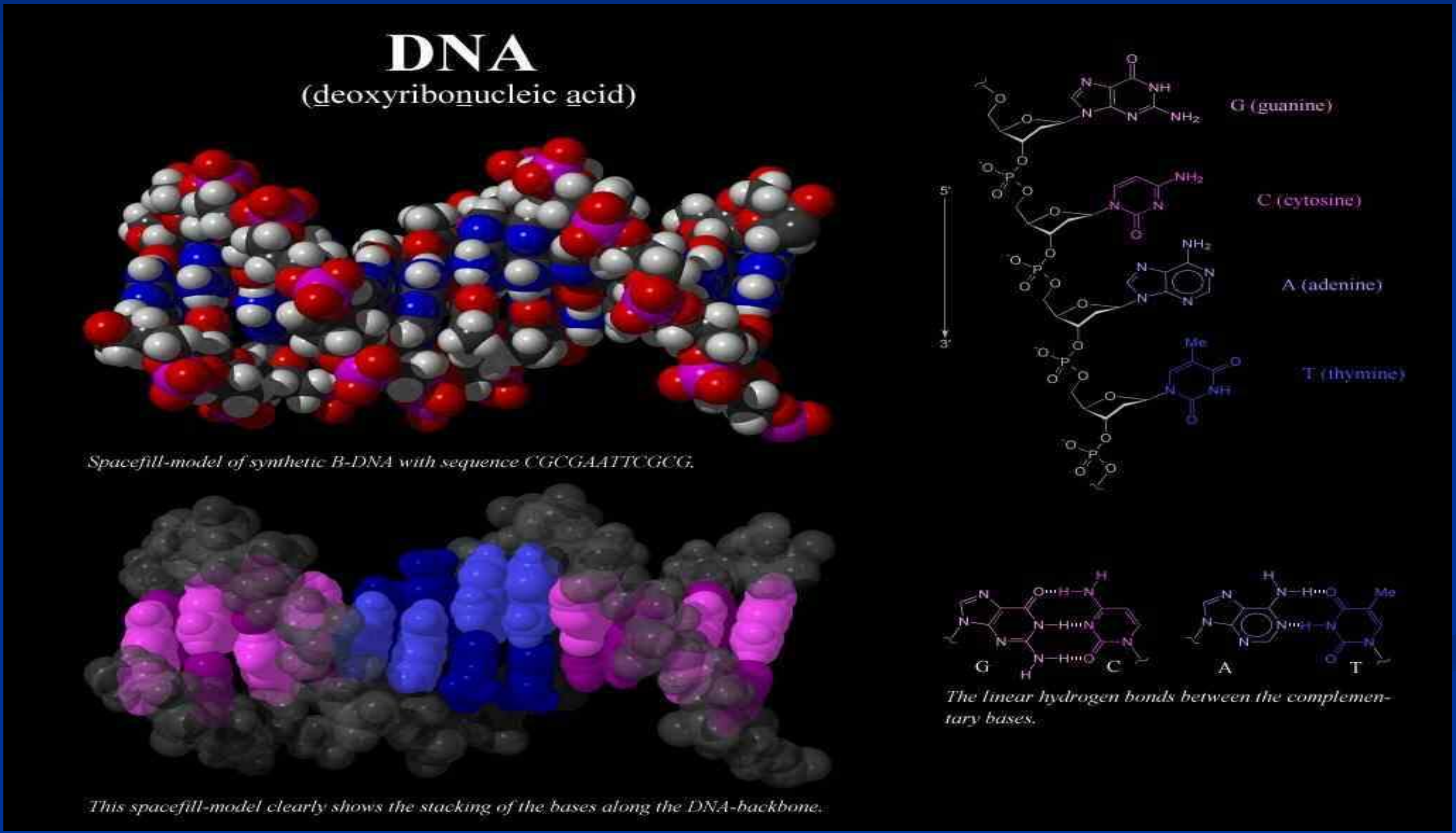
Adleman's First Demonstration of DNA Computing

Adapted from PPT of Thierry Metais

& Jaeyeon Jung and Jaehong Lim

Introduction to DNA:



Computing with DNA

- Invented (discovered?) by Dr. Leonard M. Adleman of USC in 1994, a computer scientist and mathematician
- *Basic Idea*: Perform **molecular biology** experiment to find solution to **hard** problem.
- Use "Molecular Computer" (rather than using a conventional computer for solving "computational biology" problems)

Introduction:

- What is DNA computing ?
  - Around 1950 first idea (precursor Feynman)
  - First important experiment 1994: Leonard Adleman
- Molecular level (just greater than 10^{-9} meter)
- Massive parallelism.
  - In a liter of water, with only 5 grams of DNA we get around 10^{21} bases !
  - Each DNA strand represents a bit-level processor !

A bit of biology

- The DNA is a double stranded molecule.
- Each strand is based on 4 bases:
  - Adenine (A)
  - Thymine (T)
  - Cytosine (C)
  - Guanine (G)
- Those bases are linked through a sugar (desoxyribose)
- IMPORTANT:
  - The linkage between bases has a **direction**.
  - There are **complementarities** between bases (Watson-Crick).

(A)←→ (T)

(C)←→(G)

DNA manipulations:

- If we want to use DNA as an information bulk, we must be able to manipulate it .
- However we are talking of handling molecules…
- So instead of using physical processes, we would have to use natural ones (**ENZYMES)**, more effective:
  - for lengthening: **polymerases**…
  - for cutting: **nucleases** (exo/endo-nucleases)…
  - for linking: **ligases**…

- 1985: Kary Mullis invented PCR

  Thank this reaction we get millions of identical strands, and we are allowed to think of massive parallel computing.

Coding the information:

- 1994: THE Adleman's experiment.

→ Given a *directed* graph can we find an hamiltonian path (more complex than the TSP).

→ In this experiment there are 2 keywords:

massive parallelism (all possibilities are generated)

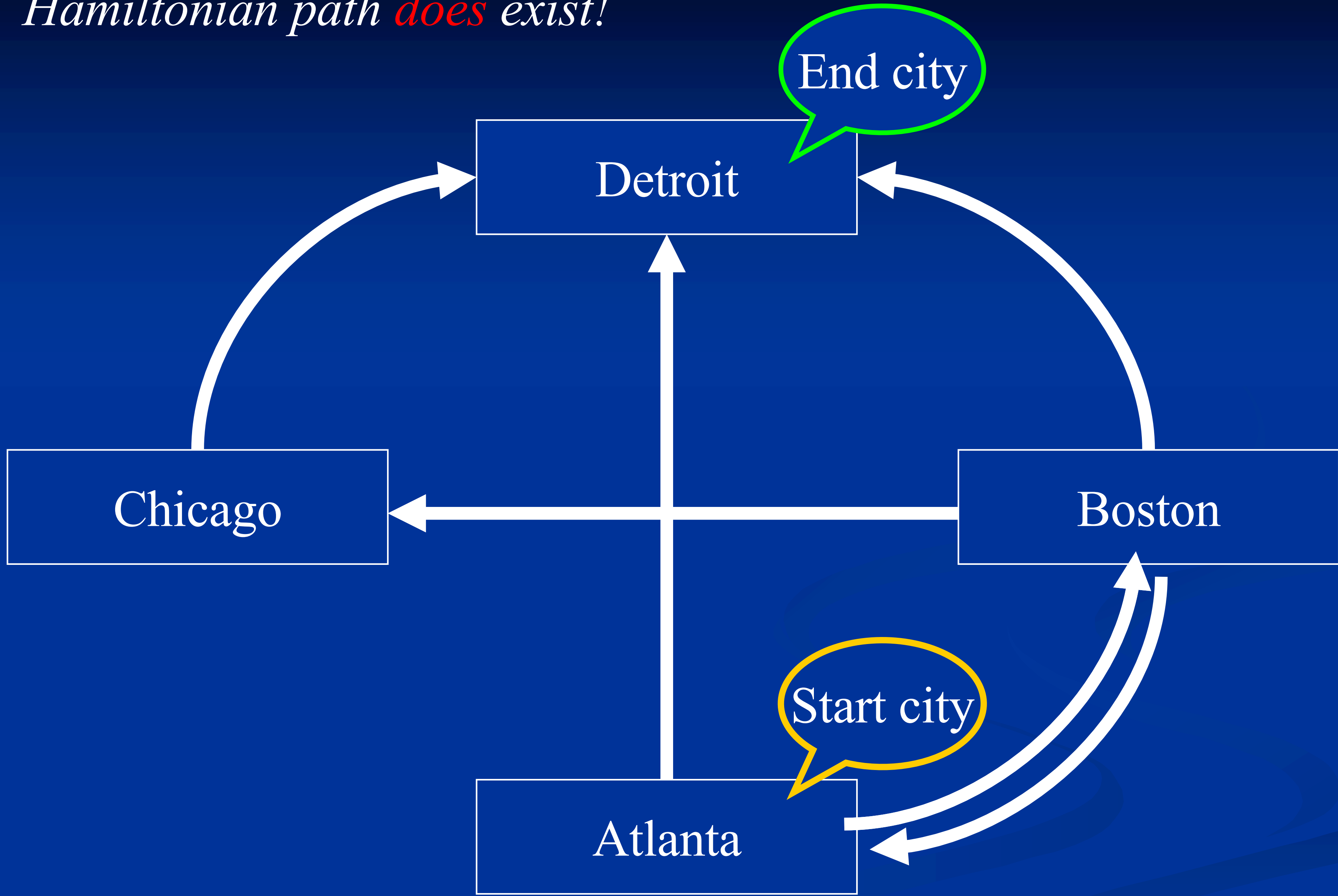
complementarity (to encode the information)

- This experiment proved that DNA computing wasn't just a theoretical study but could be applied to real problems like cryptanalysis (breaking DES).

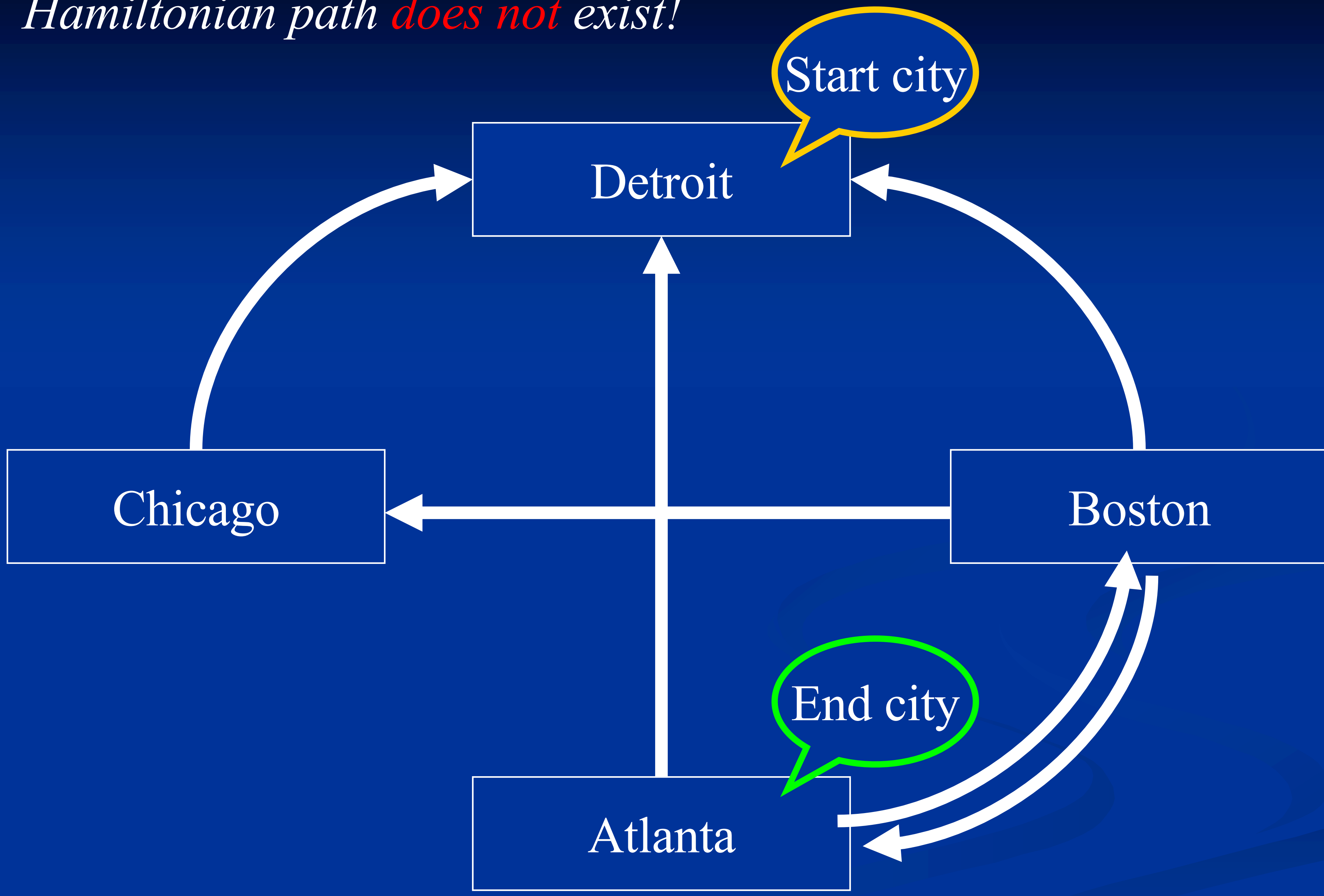
2. HAMILTONIAN PATH PROBLEM

- (Posed by William Hamilton)
- *Given a network of nodes and directed connections between them, is there a path through the network that begins with the start node and concludes with the end node visiting each node only once ("Hamiltonian path")?*
- "Does a Hamiltonian path exist, or not?"

Hamiltonian path does exist!



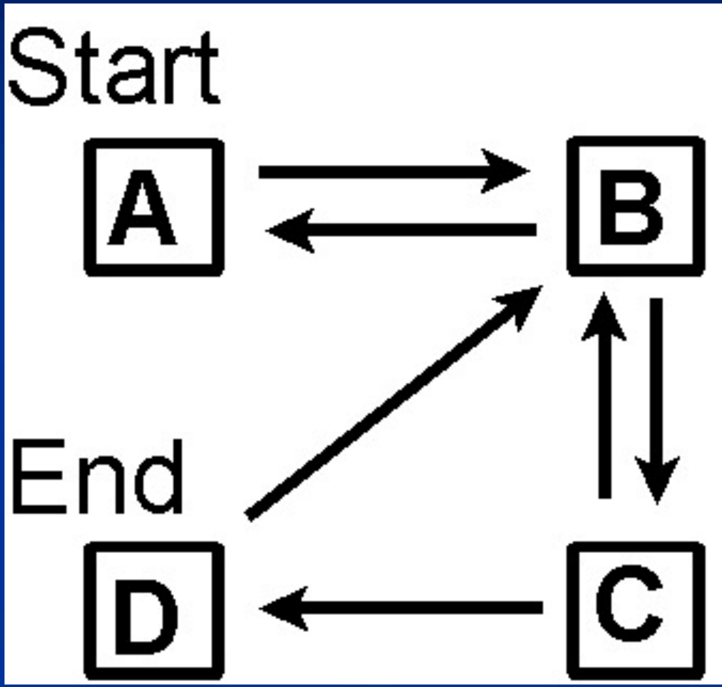
Hamiltonian path does not exist!



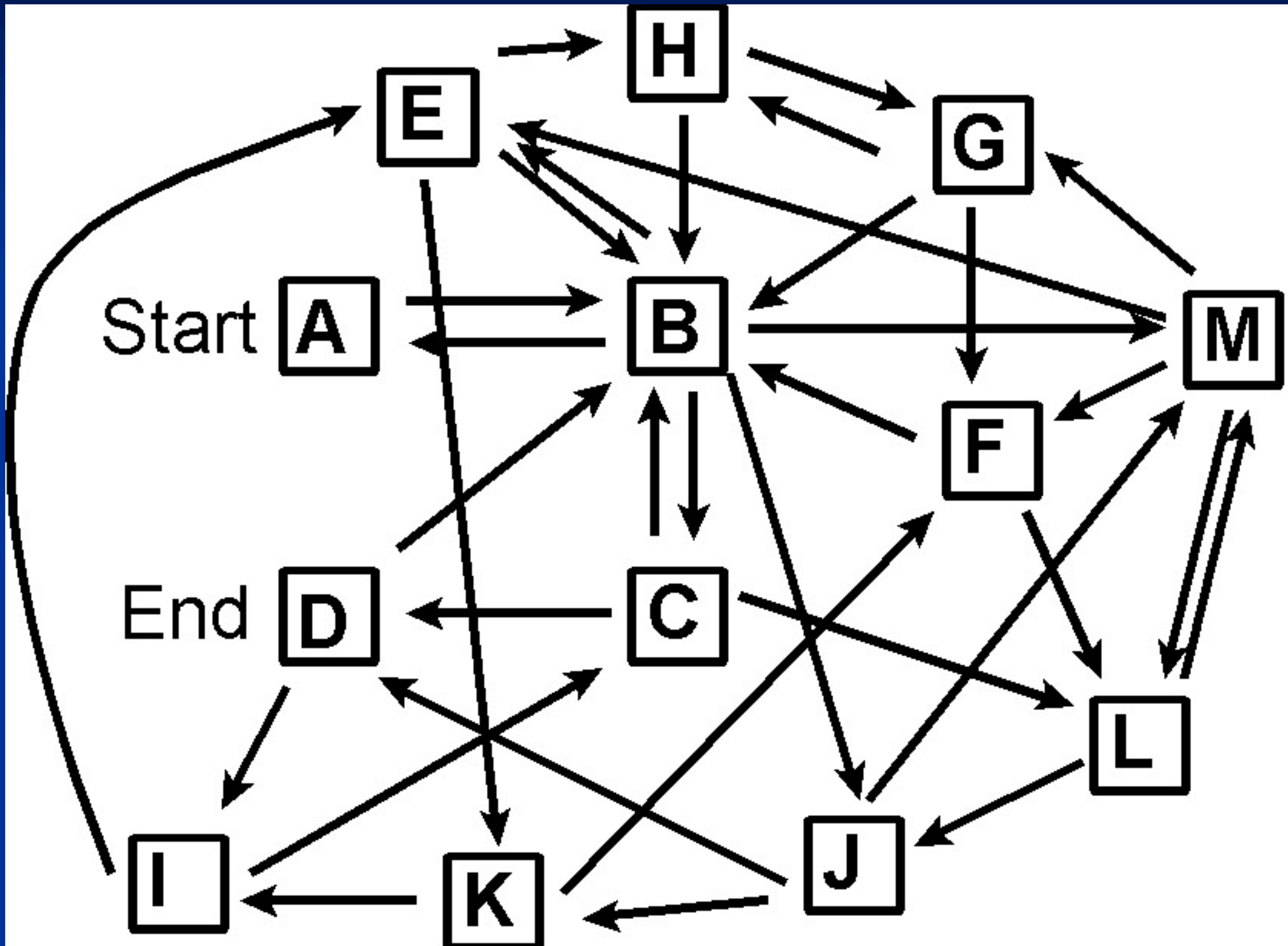
Solving the Hamiltonian Problem

- **Generation-&-Test Algorithm:**
- Step 1: Generate random paths on the network.
- Step 2: Keep only those paths that begin with start city and conclude with end city.
- Step 3: If there are N cities, keep only those paths of length N.
- Step 4: Keep only those that enter all cities at least once.
- Step 5. Any remaining paths are solutions (I.e., Hamiltonian

- **[X]** D -> B -> A
- **[X]** B -> C -> D -> B -> A -> B
- **[X]** A -> B -> C -> B
- **[X]** C -> D -> B -> A
- **[X]** A -> B -> A -> D
- **[O]** A -> B -> C -> D



Does a Hamiltonian path exist for the following network?



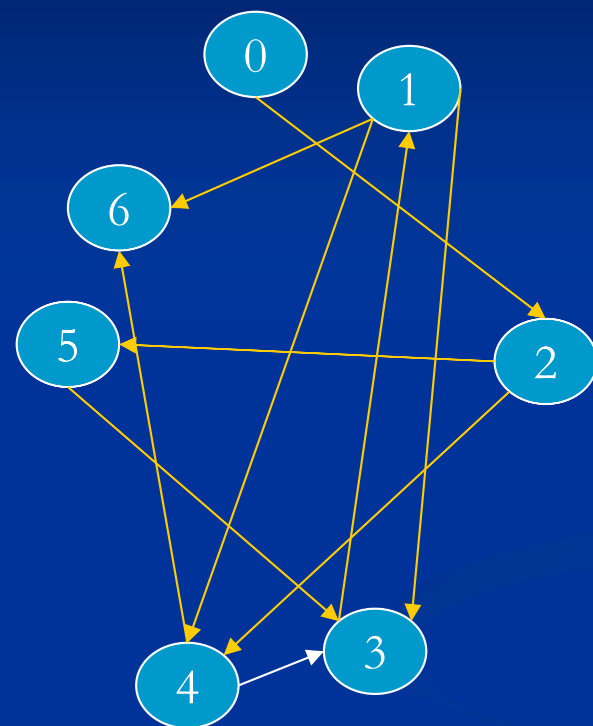
Combinatorial Explosion

- The Hamiltonian Problem is NP-hard, and
- The total number of paths grows exponentially as the network size increases
- For example:
  - 10^6 paths for N=10 cities,
  - 10^{12} paths (N=20),
  - 10^{100} paths!! (N =100)
- The *Generation-&-Test* algorithm takes "forever". Some sort of smart algorithm must be devised; none has been found so far (*NP-hard*).

Adleman experiment:

- Each node is coded randomly with 20 bases.
- Let S_i be a code, h be the complementarity mapping.
  h(ATCG) = TAGC.
- Each S_i is decomposed into 2 sub strands of length 10:
  $$S_i = S_i' S_i''$$
- Edge(i,j) will be encode as $h(S_i''S_j') \rightarrow$ (preserve edge orientation).
- **Code**:
  - Input(N) //All vertices and edges are mixed, *Nature is working*
  - $N \leftarrow B(N,S_0)$ //S_0 was chosen as input vertex.
  - $N \leftarrow E(N,S_4)$ //S_4 was chosen as output vertex.
  - $N \leftarrow E(N,<=140)$ // due to the size of the coding.
  - For i=1 to 5 do $N \leftarrow +N(N, S_i)$ //Testing if Hamiltonian path
  - Detect(N) //conclusion …

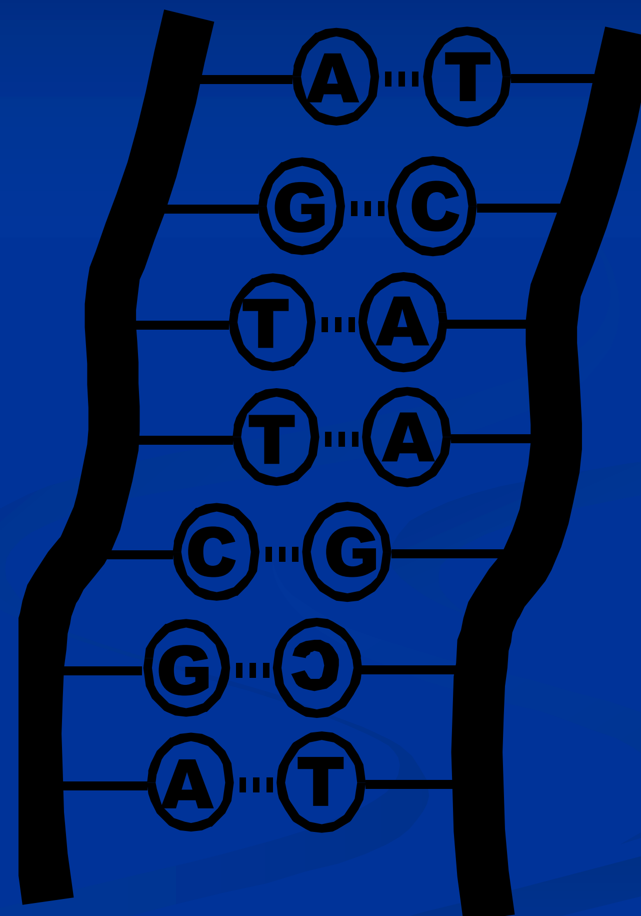
Example:



S_0	S_2	S_5	S_3	S_1	S_4	S_6
E0-2	E2-5	E5-3	E3-1	E1-4	E4-6	

3. FINDING SOLUTION WITH DNA EXPERIMENT

- DNA is a double-strand polymer made up of alternating series of four bases, A, T, C, G.
- DNA makes multiple copies of itself during cell differentiation.



DNA for Hamiltonian Problem

- The key to solving the problem is using DNA to perform the five steps of the Generation-&-Test algorithm in **parallel search**, instead of serial search.

Solving the Hamiltonian Problem

Generation-Test Algorithm:

Step 1: Generate random paths on the network.

Step 2: Keep only those paths that begin with the start city and conclude with the end city.

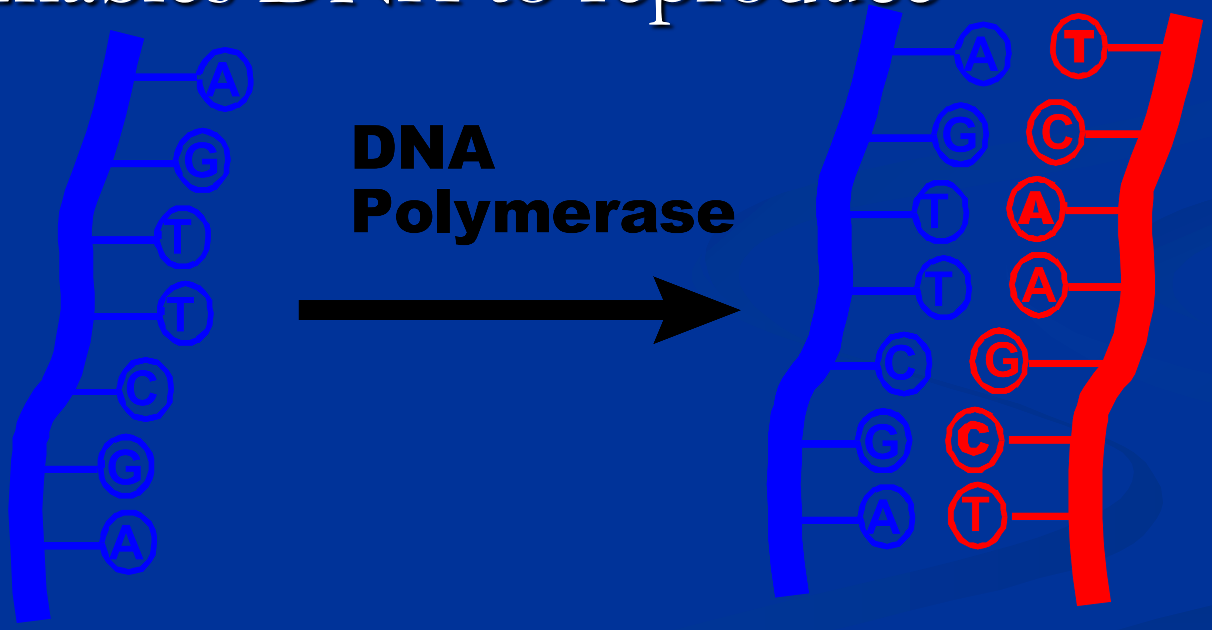
Step 3: If there are N cities, keep only those paths of length N.

Step 4: Keep only those paths that enter all cities at least once.

Step 5. Any remaining paths are solutions.

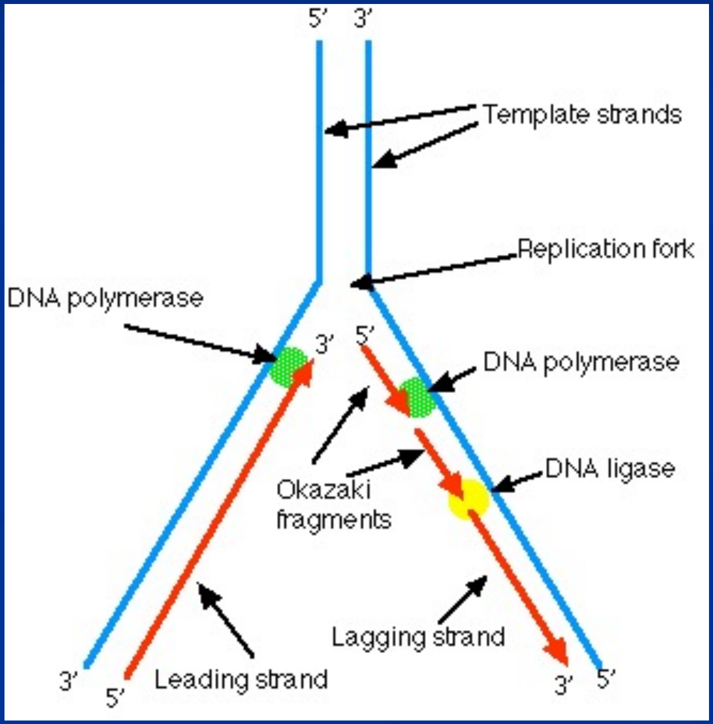
DNA Polymerase

- - Protein that produces complementary DNA strand
- - A -> T, T -> A, C -> G, G -> C
- - Enables DNA to reproduce



Polymerase in Action

- The "Bio" nano-machine:
  - *hops* onto DNA strand
  - *slides* along
  - *reads* each base
  - *writes* its complement onto new strand

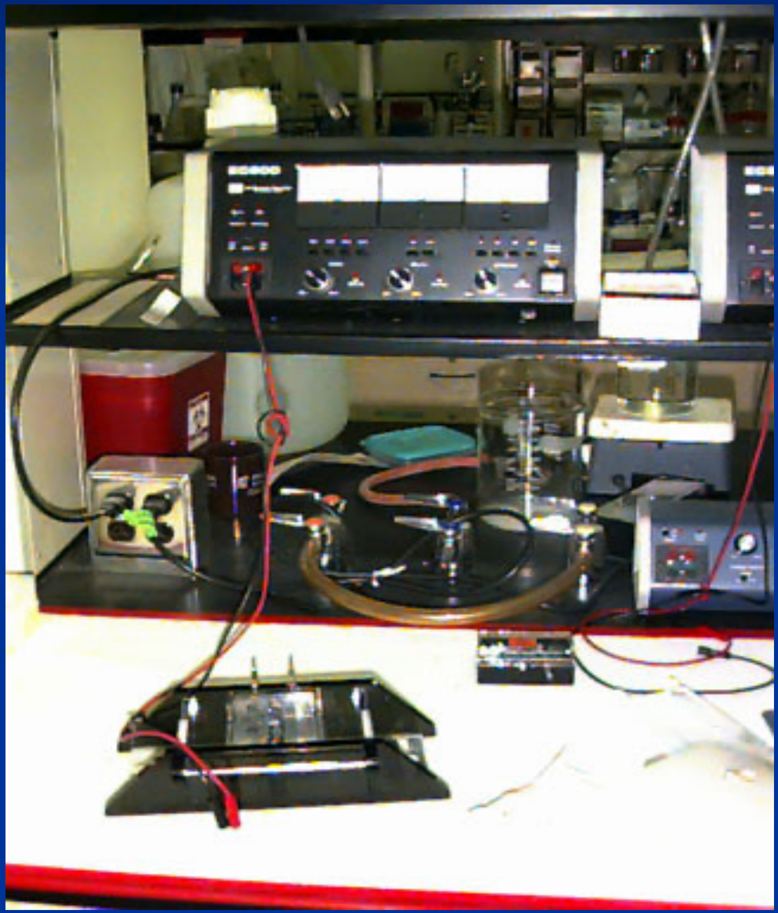


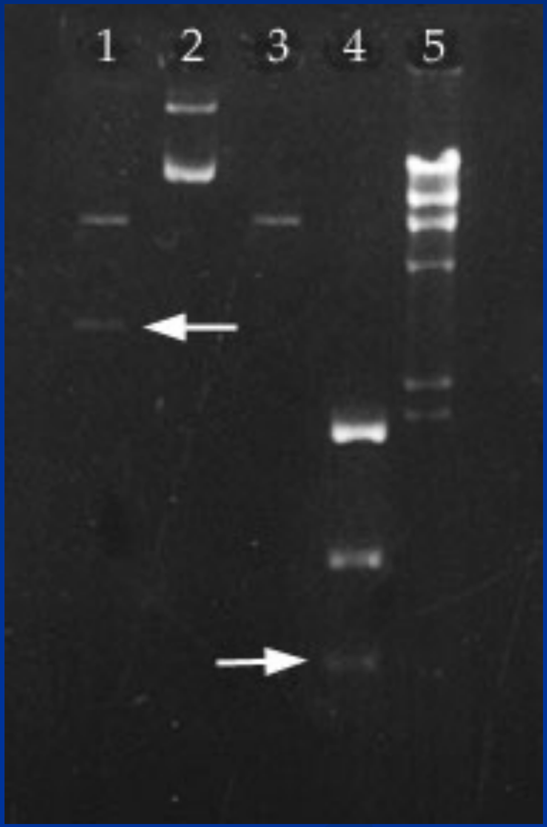
DNA Experiment Set-up

Ingredients and tools needed:

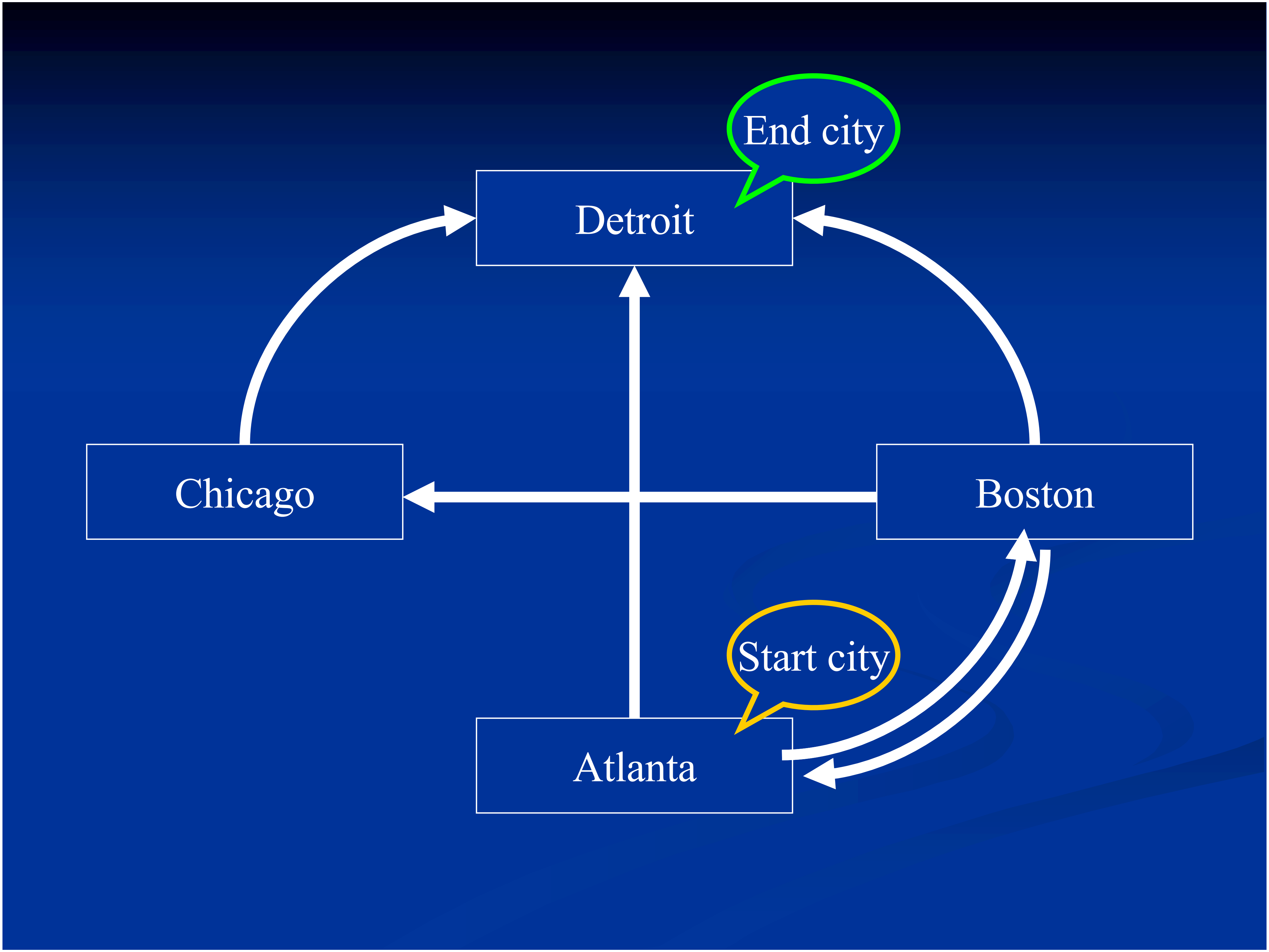
- DNA strands that encode city names and connections between them
- Ligase, water, salt, other ingredients
- Polymerase chain reaction (PCR) set
- Gel electrophoresis tool (that filters out non-solution strands)

Gel Electrophoresis





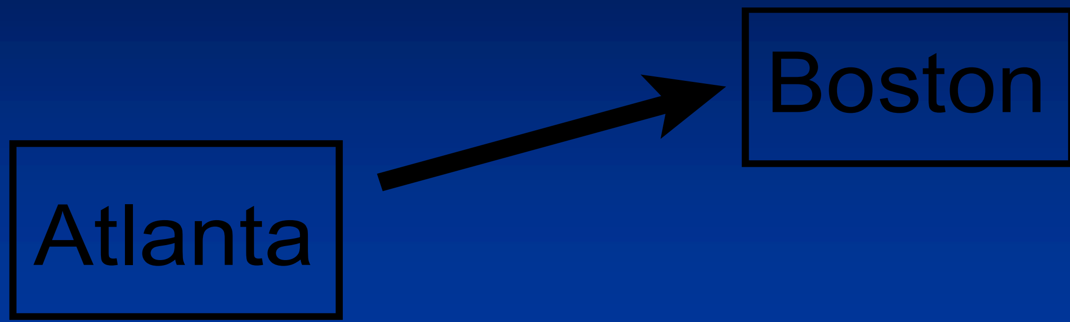
http://www.life.uiuc.edu/molbio/geldigest/equipment.html


End city
Detroit
Chicago
Boston
Start city
Atlanta

CITY	DNA NAME	COMPLEMENT
ATLANTA	ACTT*GCAG*	TGAA*CGTC*
BOSTON	TCGG*ACTG*	AGCC*TGAC*
CHICAGO	GGCT*ATGT*	CCGA*TACA*
DETROIT	CCGA*GCAA*	GGCT*CGTT*

CONNECTING PATH	DNA PATH
ATLANTA-BOSTON	*GCAG*TCGG
ATLANTA-DETROIT	*GCAG*CCGA
BOSTON-CHICAGO	*ACTG*GGCT
BOSTON-DETROIT	*ACTG*CCGA
BOSTON-ATLANTA	*ACTG*ACTT
CHICAGO-DETROIT	*ATGT*CCGA

DNA encoding of city-network

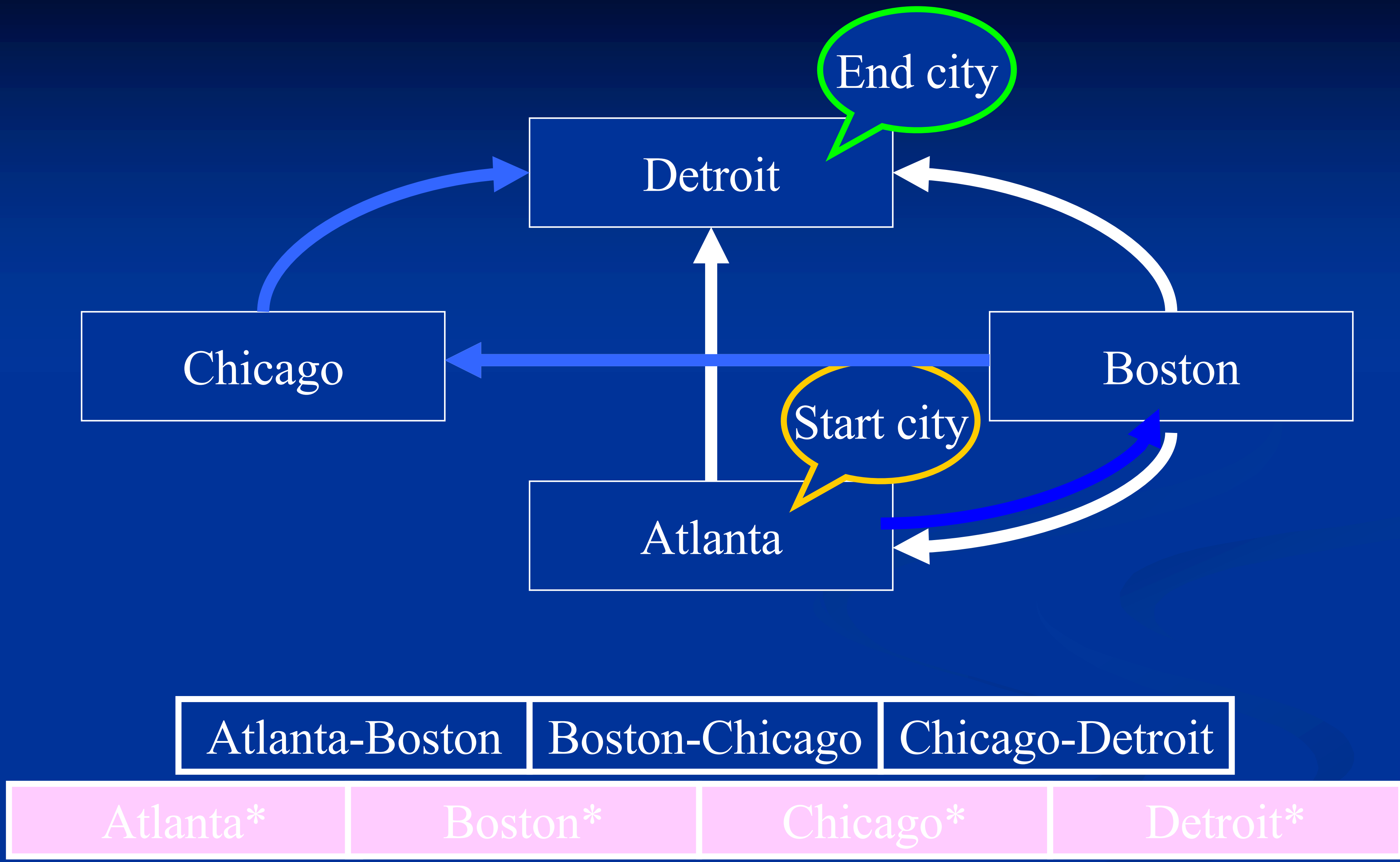


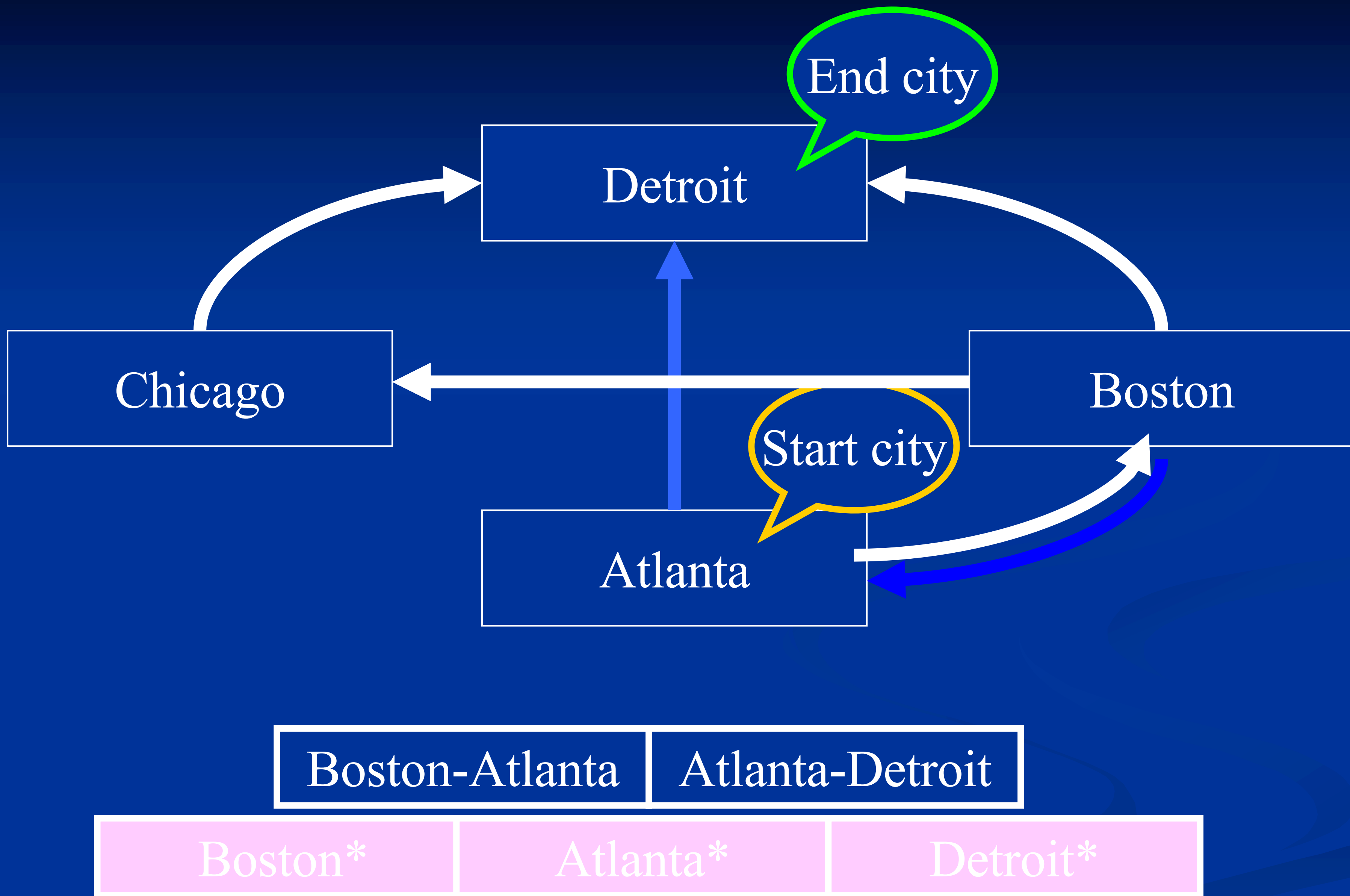
Atlanta -Boston

GCAGTCGG

TGAACGTC **AGCCTGAC**

Atlanta Boston


End city
Detroit
Chicago
Boston
Start city
Atlanta
Atlanta-Boston
Boston-Chicago
Chicago-Detroit
Atlanta*
Boston*
Chicago*
Detroit*


End city
Detroit
Chicago
Boston
Start city
Atlanta
Boston-Atlanta
Atlanta-Detroit
Boston*
Atlanta*
Detroit*

Adleman's DNA Experiment

- 1. In a test tube, mix the prepared DNA pieces together
  - Which will randomly link with each other, forming all different paths.
  - Ligase will heal nicks between consecutive cities, allowing each path to be a DNA strand (representing a possible Hamiltonian path) .

Adleman's DNA Experiment

- 2. Perform PCR with two 'start' and 'end' DNA pieces as primers
  - Which creates many copies of each DNA strand (representing a possible Hamiltonian path) with the correct start and end.

Adleman's DNA Experiment

- 3. Perform gel electrophoresis to identify only those pieces of right length (e.g., N=4).

- 4. For each city:
  - Use DNA-attached magnetic probe sepearation to separate out the DNA sequences that contain that city.
  - These magnetic probes are magnetic nanoparticles with an attached DNA strand that is complementary to the given city.
  - Discard the DNA sequences that do not contain that city.

- 5. All DNA pieces that are left in the final test tube should be precisely those representing Hamiltonian paths.
  - If the final test tube contains any DNA at all, then conclude that a Hamiltonian path exists, and otherwise not.
  - When it does, the DNA sequence represents the specific path of the solution.

4. SUMMARY & CONCLUSION

- **Enormous parallelism**,
  - with 10^{23} DNA pieces working in parallel to find solution simultaneously.
  - Takes less than a week (vs. thousands yrs for supercomputer)
- **Extraordinary energy efficient**
  - (10^{-10} of supercomputer energy use)
- **But limited by exponential size growth of amount of DNA needed**

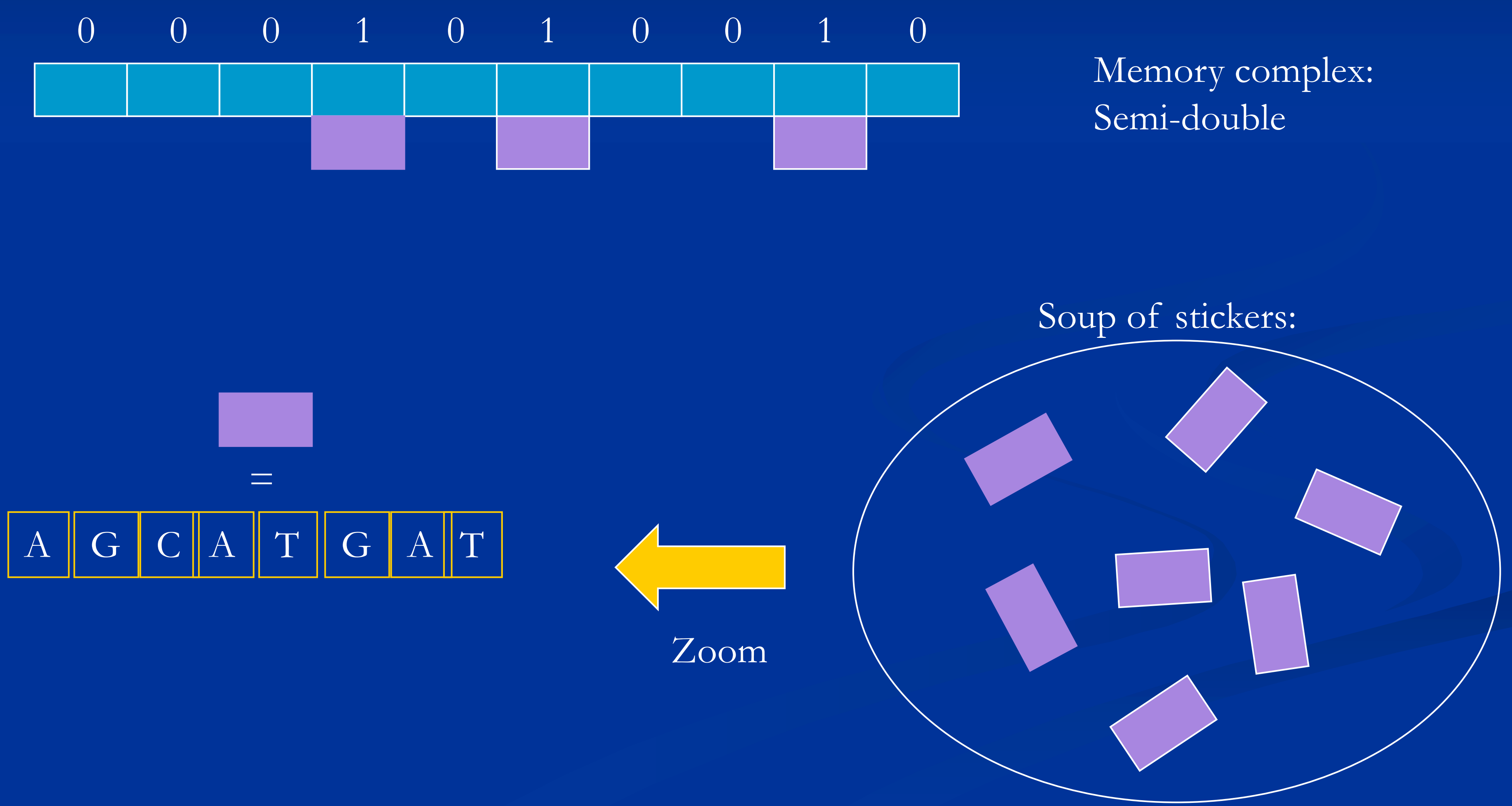
New generation of computers?

- In the second part of [1], it is proven through language theory that DNA computing "guarantees universal computations".
- Many architectures have been invented for DNA computations.
- The Adleman experiment is not the single application case of DNA computing…

Stickers model:

- Memory complex = Strand of DNA (single or semi-double).
- Stickers are segments of DNA, that are composed of a certain number of DNA bases.
- To use correctly the stickers model, each sticker must be able to anneal only at a specific place in the memory complex.

To visualize:



About a stickers machine?

- Simple operations: merge, select, detect, clean.
- → Tubes are considered (cylinders with two entries)
- However for a mere computation (DES):
  - Great number of tubes is needed (1000).
  - Huge amount of DNA needed as well.
- Practically no such machine has been created….

  → Too much engineering issues.

Why don't we see DNA computers everywhere?

- DNA computing has wonderful possibilities:
  - Reducing the time of computations* (parallelism)
  - Dynamic programming !
- However one important issue is to find "the killer application".
- Great hurdles to overcome…

Some hurdles:

- Operations done manually in the lab.
- Natural tools are what they are…
  - →Formation of a library (statistic way)
  - →Operations problems

Conclusion:

- The paradigm of DNA computing has lead to a very important theoretical research.
- However DNA computers won't flourish soon in our daily environment due to the technologic issues.
- Adleman renouncement toward electronic computing.
- Is all this work lost ?
- NO ! → "Wet computing"