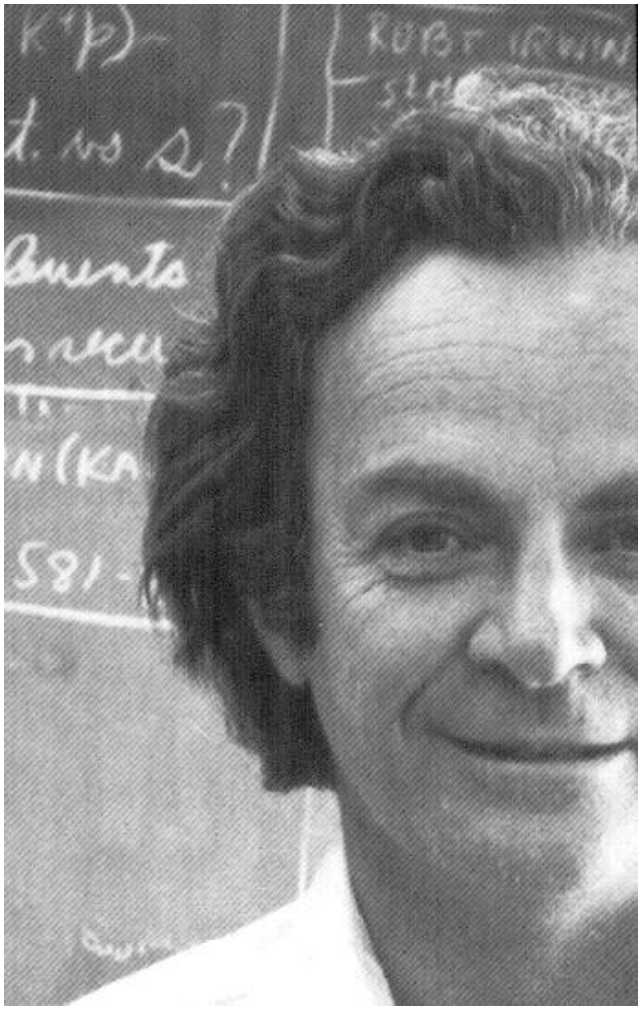
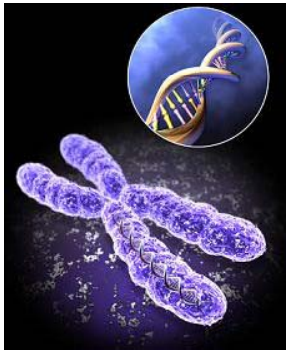
DNA Computing

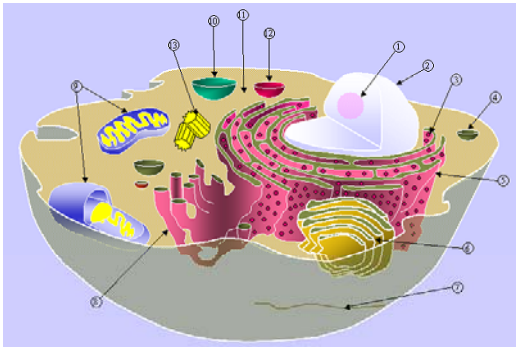
Computing on molecular level



- Richard P. Feynman
- In late 1950s,

- Biological molecules can carry enormous amounts of information in an exceedingly small space.
  → inborn computing power!





Motivation of DNA Computing

- We need a totally different new technology to overcome CMOS limitations.
- Certain types of problems (learning, pattern recognition, large set search algorithms) are intrinsically very difficult to solve even with fast evolution of CMOS.
- It is natural to solve biological problems with biological tools.

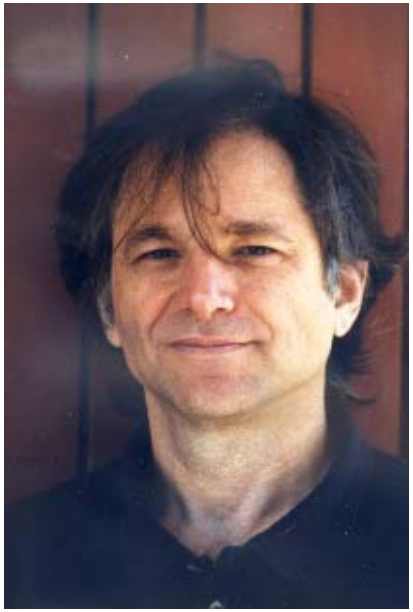
Advantages of DNA Computing

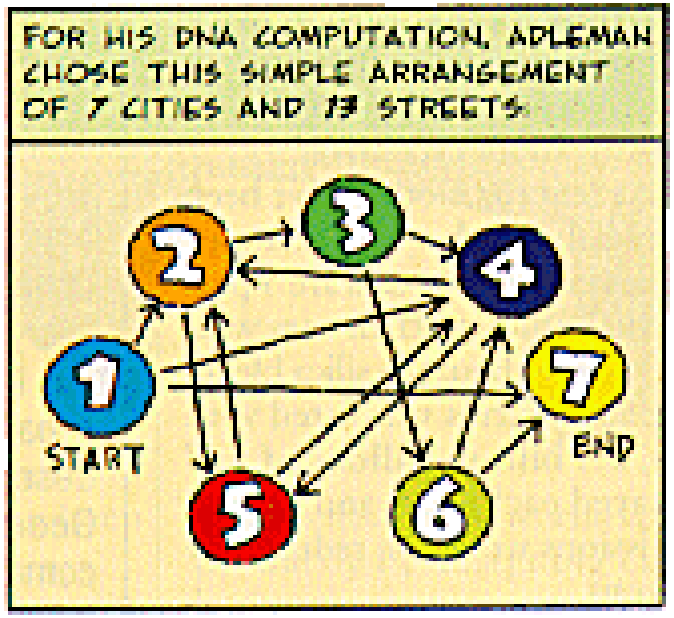
- Achievement of massive parallelism
- Parallel molecular operation
  - Desktop: 10^9 operations/ sec
  - Supercomputer: 10^{12} operations/ sec
  - 1 mole of DNA: 10^{23} simultaneous reactions
- High information storage capacity
  - 6.022 *10^{23} molecules/ mole → 1 bit per cubic nanometer
- Favorable Energetics

Rise and Growth of DNA Computing

- Adleman's work in 1994
  - Hamiltonian path problem (graph problem, NP problem)
  - City and road information representation using DNA sequences (indicative information)
  - Solution-based DNA computing
- Liu's work in 2000
  - SAT problem
  - Surface-based DNA computing
- Benenson's work in 2004
  - Application to disease diagnosis and drug (antisense) administration

Rise of DNA computing

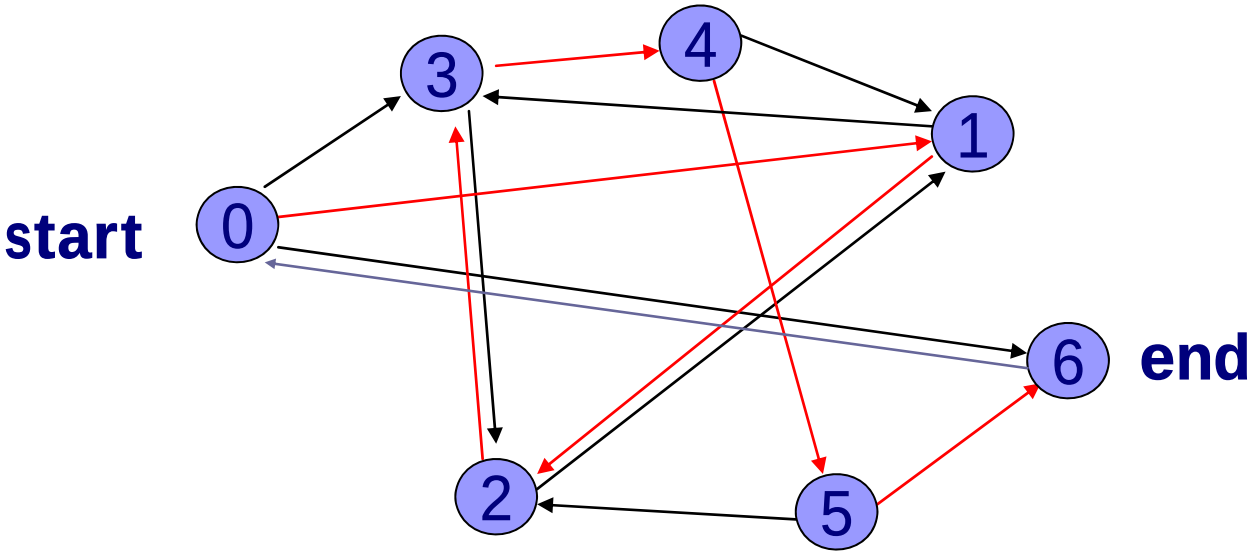




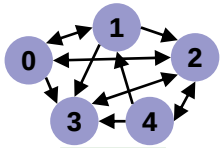
- In 1994, Adleman showed that DNA ‘can compute’.
- Hamiltonian path problem (HPP)

Hamiltonian Path Problem (HPP)

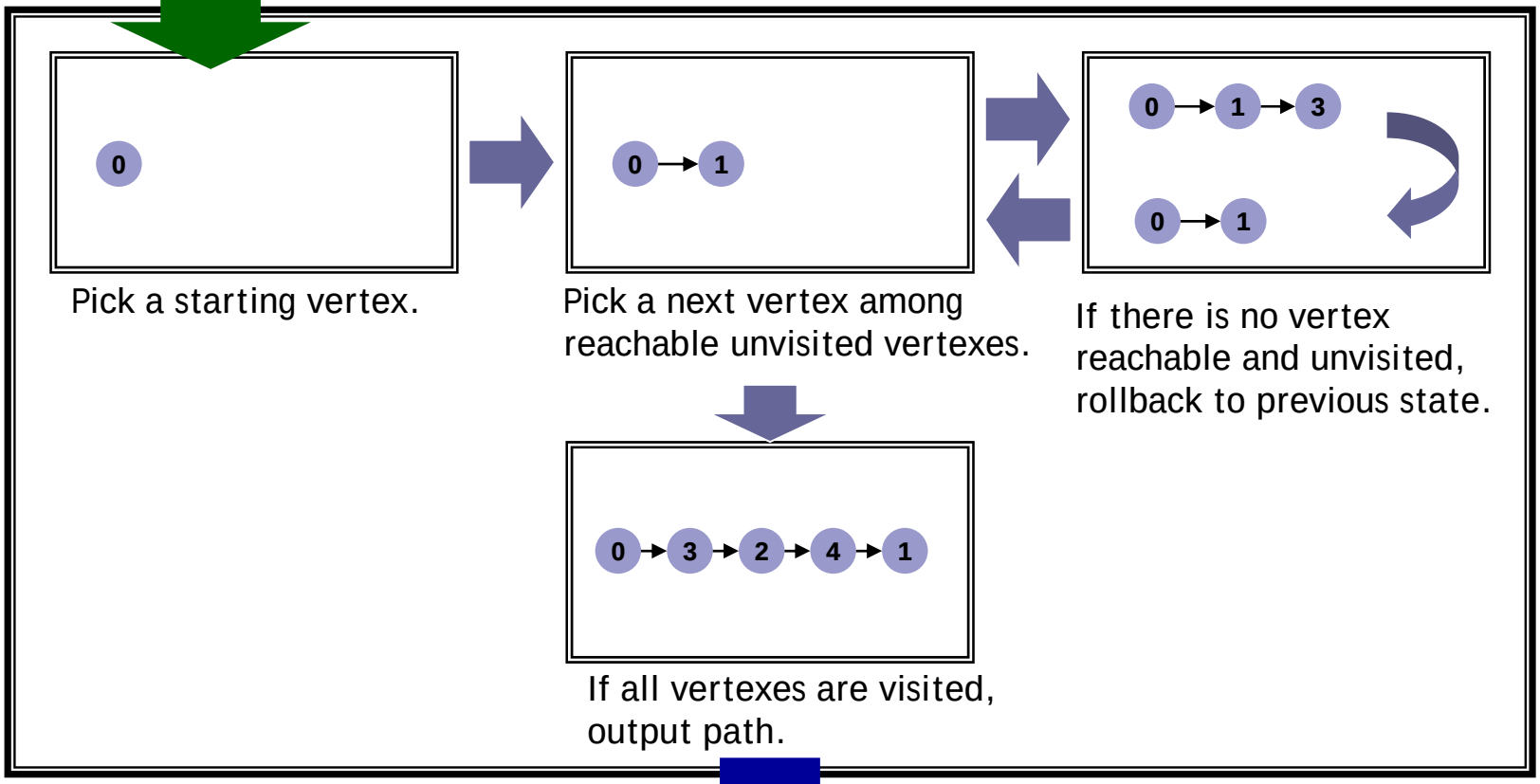
- HPP is to find a route (if it exists) that passes through each city exactly once with a designated start and end.



Conventional computers



Input:
Graph information: vertexes and edges



Pick a starting vertex.

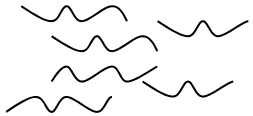
Pick a next vertex among reachable unvisited vertexes.

If there is no vertex reachable and unvisited, rollback to previous state.

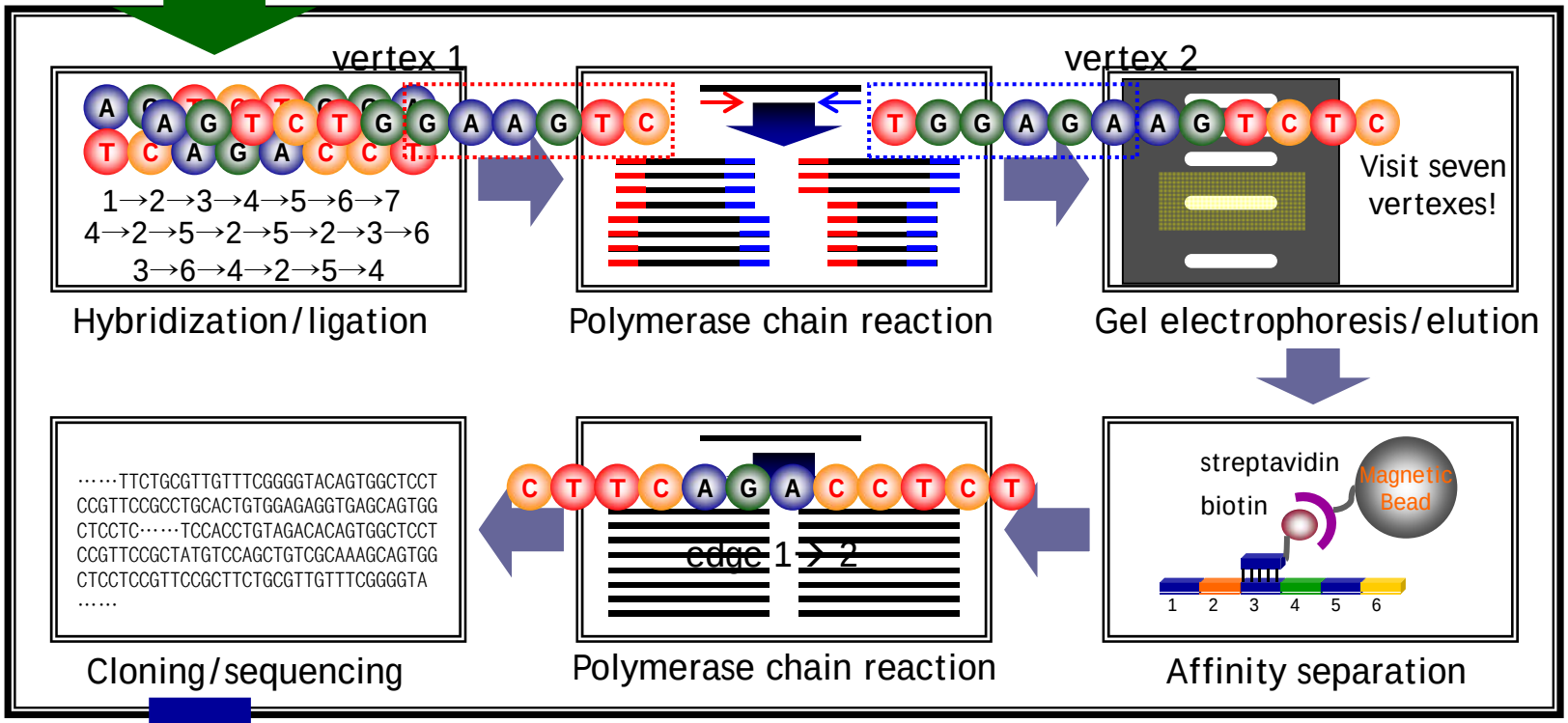
If all vertexes are visited, output path.

Output:
Hamiltonian path

DNA computers for HPP



Input:
Oligonucleotides mixture
(vertexes, edges)



Output:
DNA sequence
of Hamiltonian path

Experimental Implementation for HHP Conditions

- Starting with city 0 and ending with city 6
  - PCR using primers complementary to city 0 and city 6
- Visiting seven cities
  - Gel electrophoresis/elution
- Visiting every city
  - A series of affinity chromatography
  - Each affinity column contains ssDNA complementary to each city.

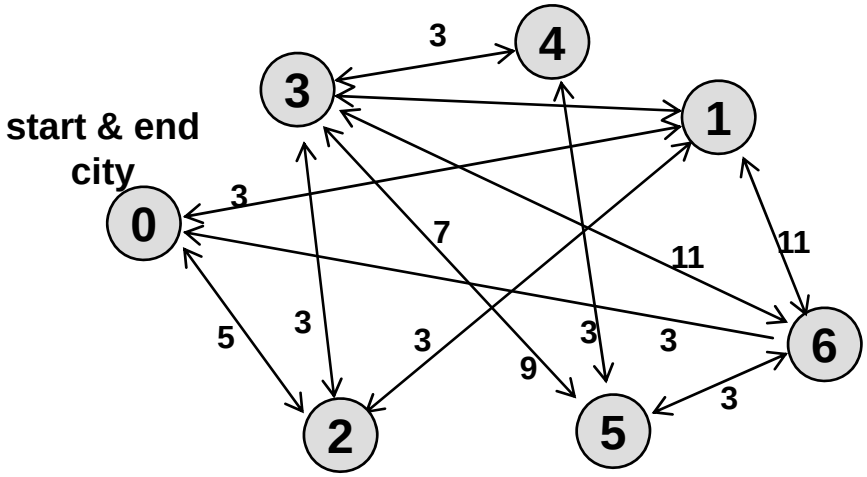
Traveling Salesman Problem

Traveling Salesman Problem

- Find…
  - The cheapest way of visiting all the cities and returning to the starting point

    when a number of cities to visit and the traveling cost between each pair of cities are given.
- Previous work for weight (cost) representation
  - DNA length
  - DNA concentration
- Our method for weight (cost) representation
  - Thermal stability of DNA duplex
  - Melting temperature (T_m), GC content

Target Problem & Encoding Method

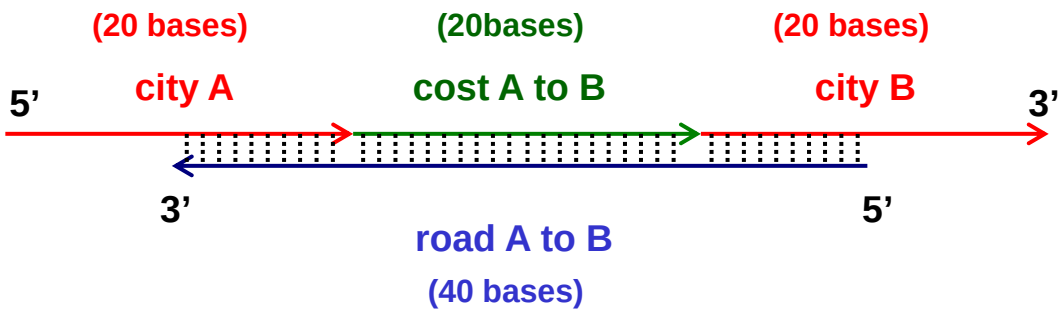


7-city traveling salesman problem

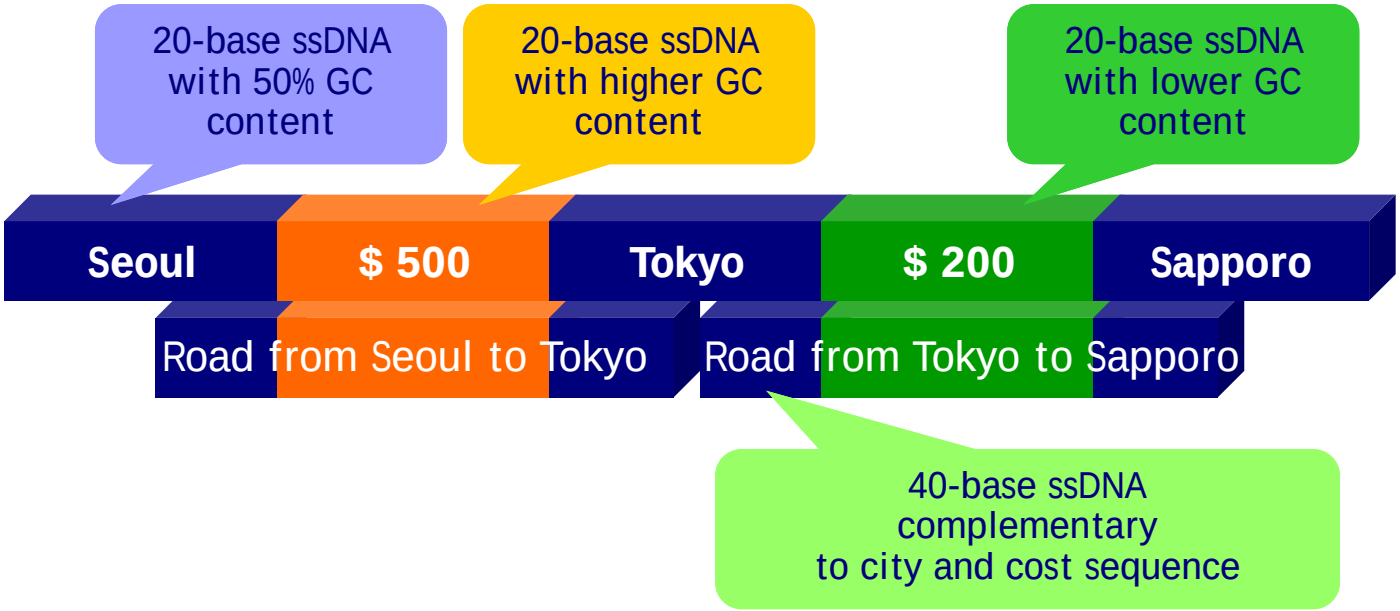
- 7 cities (0 to 6), 23 roads, 5 costs
- optimal path: '0→1→2→3→4→5→6→0'

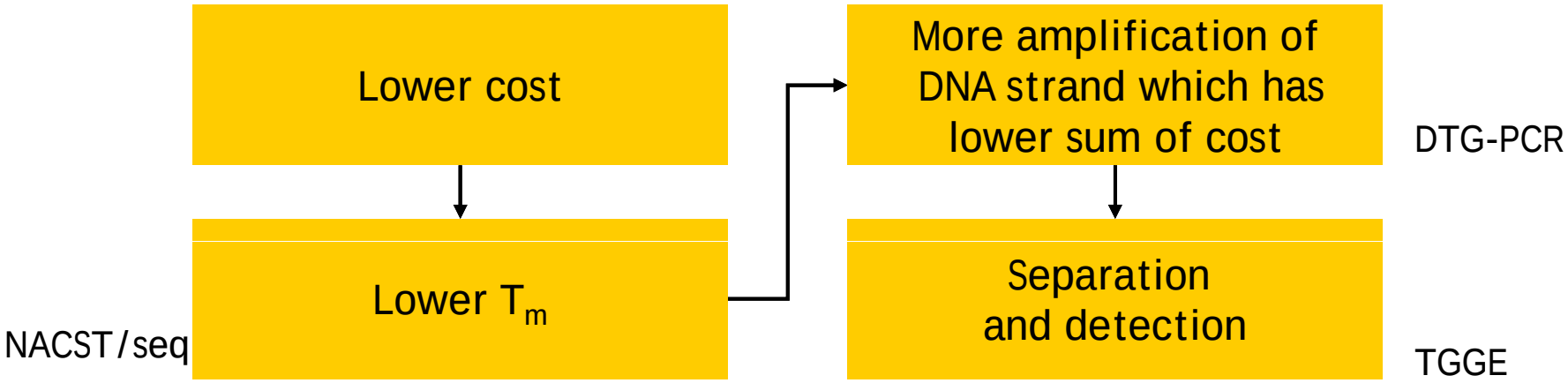
Oligonucleotides & Encoding

- cities and costs are 20-base ssDNA
- roads are 40-base ssDNA
- 35 oligonucleotides
  (7 cities, 23 roads, 5 costs)



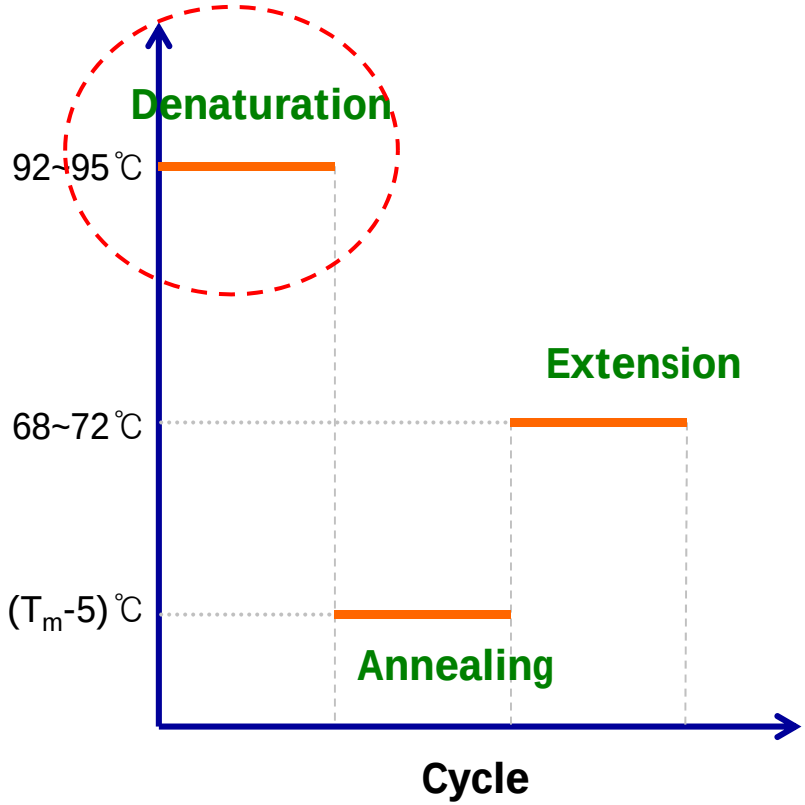
Weight (Cost) Encoding



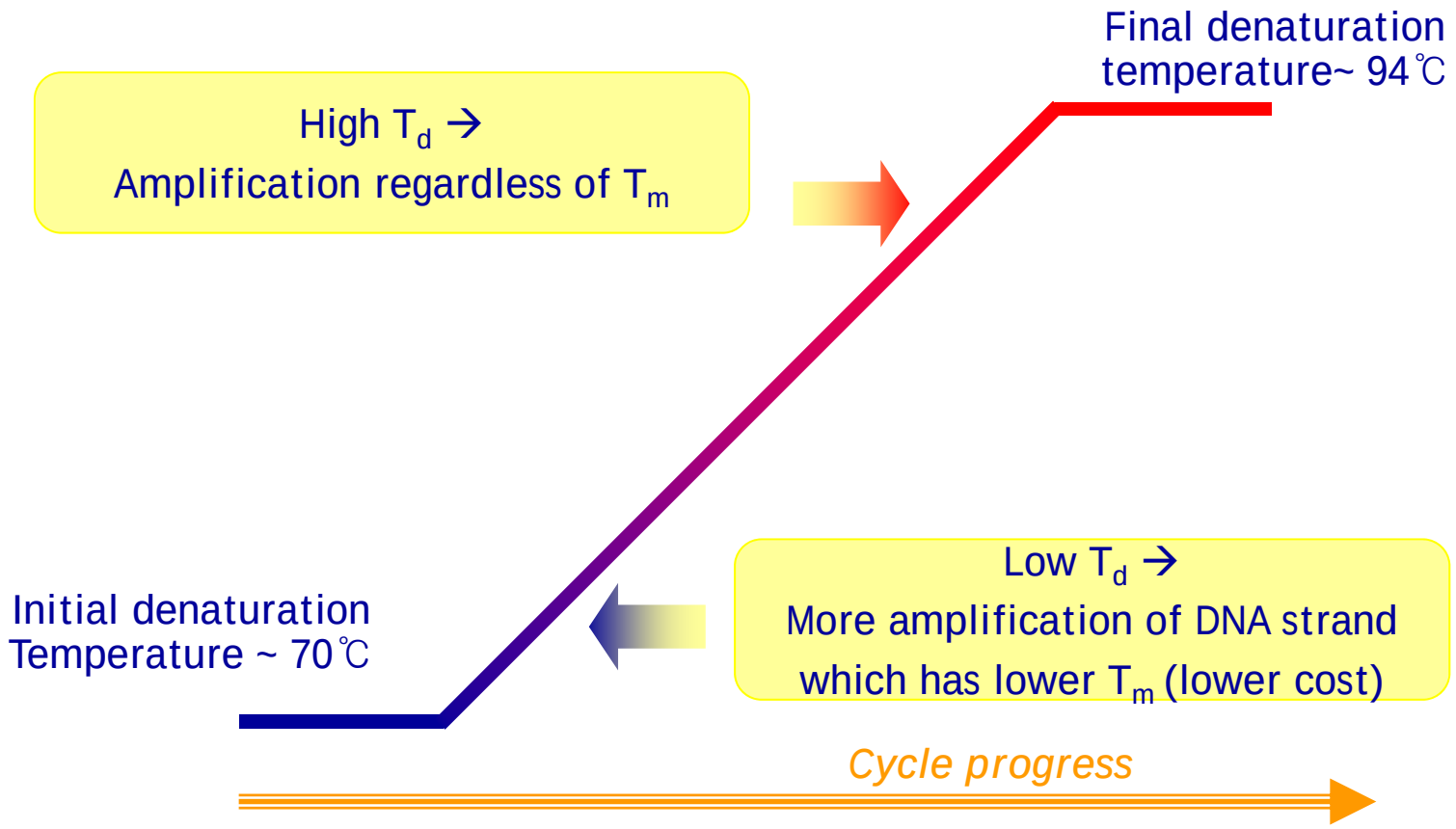


Denaturation Temperature Gradient Polymerase Chain Reaction (DTG-PCR)

- Conventional PCR
  - Denaturation (T_d)
  - Annealing (T_a)
  - Extention (T_e)
- Modification of conventional PCR protocol
  - Variation in T_d

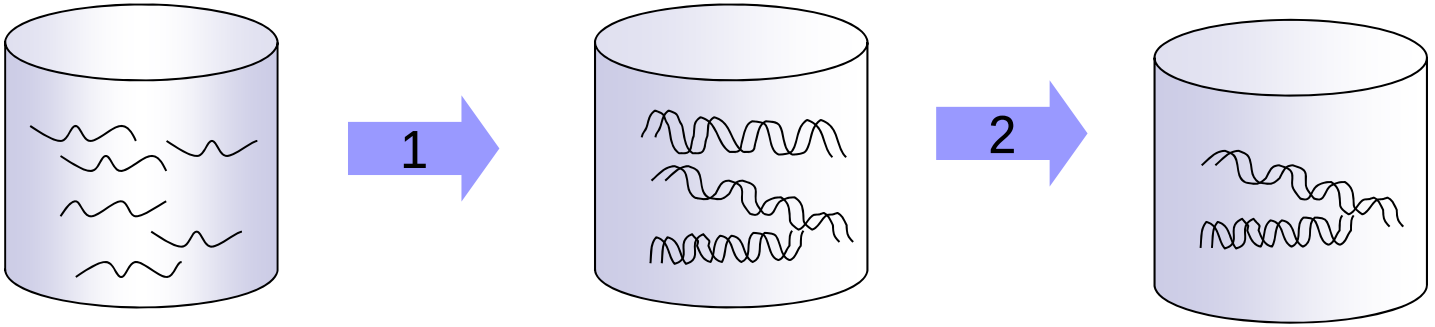


Denaturation Temperature Gradient



- Biased operator: more amplification of DNA strands with lower T_m
  → biased search for lower cost

Molecular Algorithm

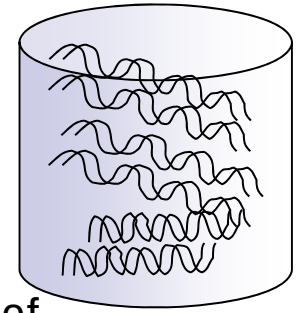


(1) Generation of all random paths
Hybridization/ligation

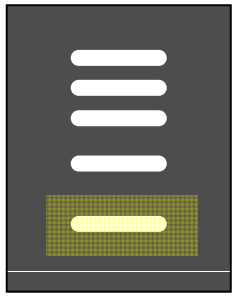
(2) Selection of paths satisfying TSP conditions
PCR/electrophoresis/affinity

3

(3) More amplification of the cheapest path
DTG-PCR



4



5

Readout!
Sequencing

(5) Elution of the cheapest path
Gel elution

(4) Separation of the cheapest path
TGGE

Sequence Design for Cities and Costs

- Using NACST/seq
- Non-cross hybridization
- Similar T_m among cities
- Different T_m among costs

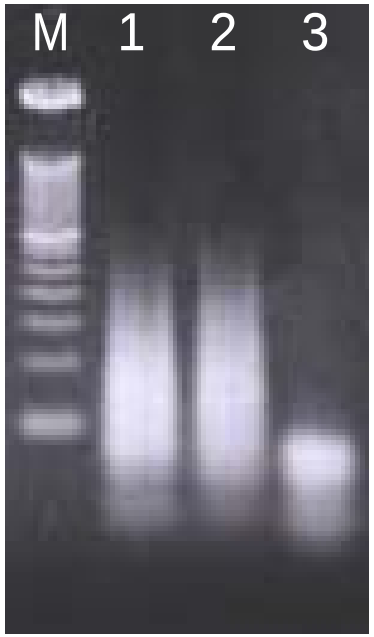
Vertex sequences			
No.	Sequence ($5' \rightarrow 3'$)	Tm	GC%
0	AGGCGAGTATGGGGTATATC	60.73	50
1	CCTGTCAACATTGACGCTCA	59.24	50
2	TTATGATTCCACTGGCGCTC	59.00	50
3	ATCGTACTCATGGTCCCTAC	56.81	50
4	CGCTCCATCCTTGATCGTTT	58.13	50
5	CTTCGCTGCTGATAACCTCA	59.44	50
6	GAGTTAGATGTCACGTCACG	56.97	50
Weight sequences			
Edge cost	Sequence ($5' \rightarrow 3'$)	Tm	GC%
3	ATGATAGATATGTAGATTCC	47.89	30
5	GGATGTGATATCGTTCTTGT	54.62	40
7	GGATTAGCAGTGCCTCAGTT	58.37	50
9	TGGCCACGAAGCCTTCCGTT	64.51	60
11	GAGCTGGCTCCTCATCGCGC	68.88	70

Experimental Implementation for TSP Conditions

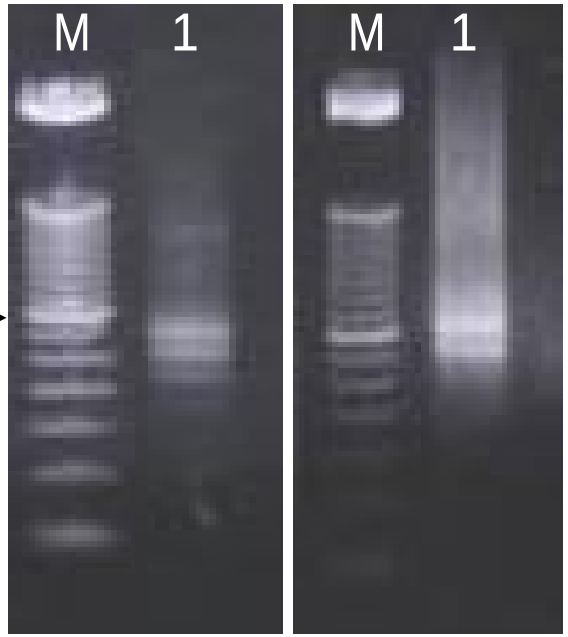
- Strating and ending with city 0
  - PCR using primers complementary to city 0
- Visiting every city
  - A series of affinity chromatography
  - Each affinity column contains ssDNA complementary to each city.
- Cheapest path
  - DTG-PCR

Experimental Results

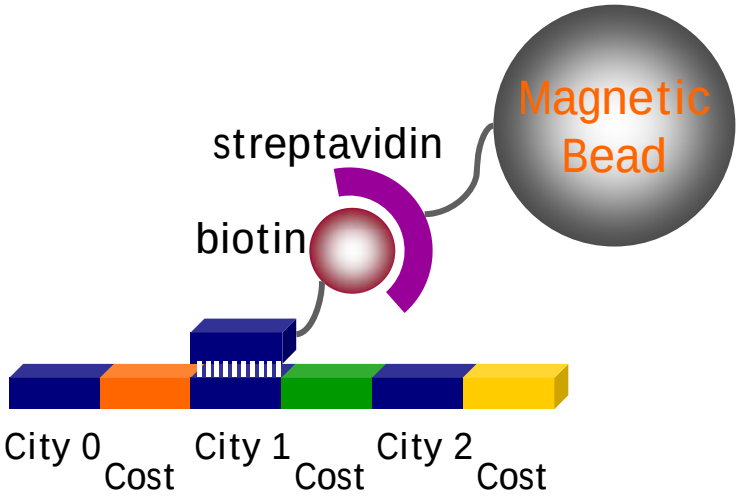
- Random path generation
  (by hybridization and ligation)
- Selective amplification of paths starting and ending with city 0
  (by PCR using primers complementary to city 0)

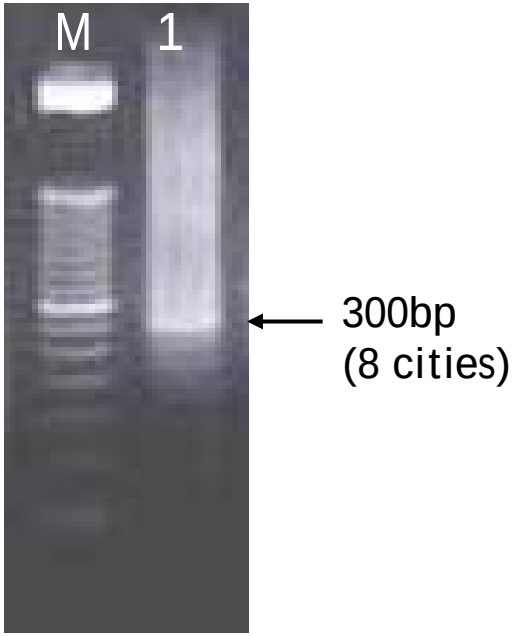


M: 50 bp ladder
lane 1,2:
after hybridization/ligation
lane 3: mixture of ssDNA

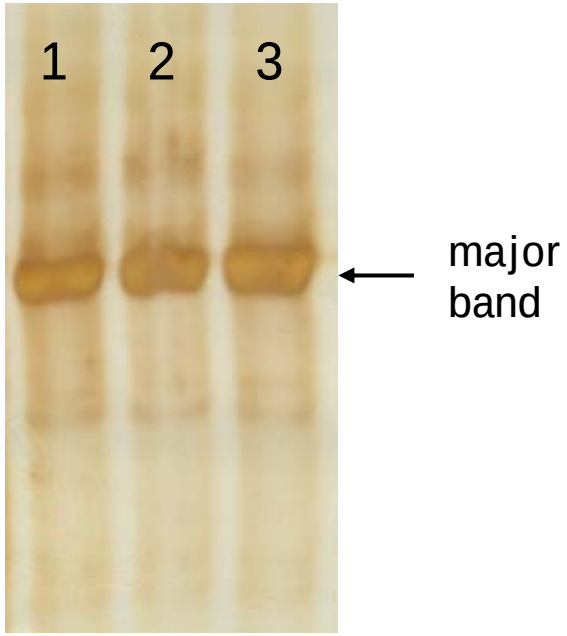


- Separation of paths containing every city
  (by a series of affinity chromatography)
- More amplification of paths with lower costs
  (by DTG-PCR)

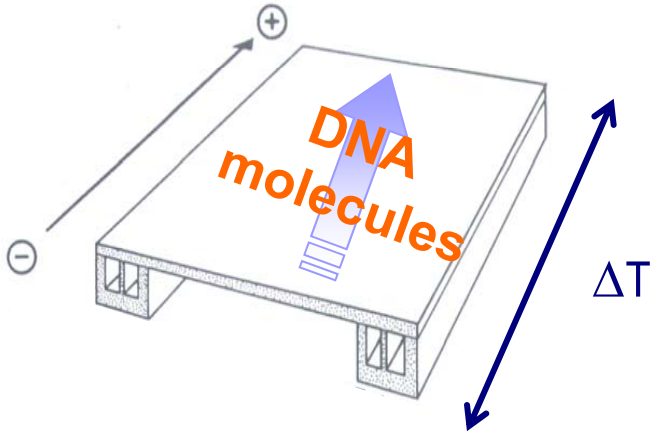




- Separation of the path with lowest cost
  (by TGGE)



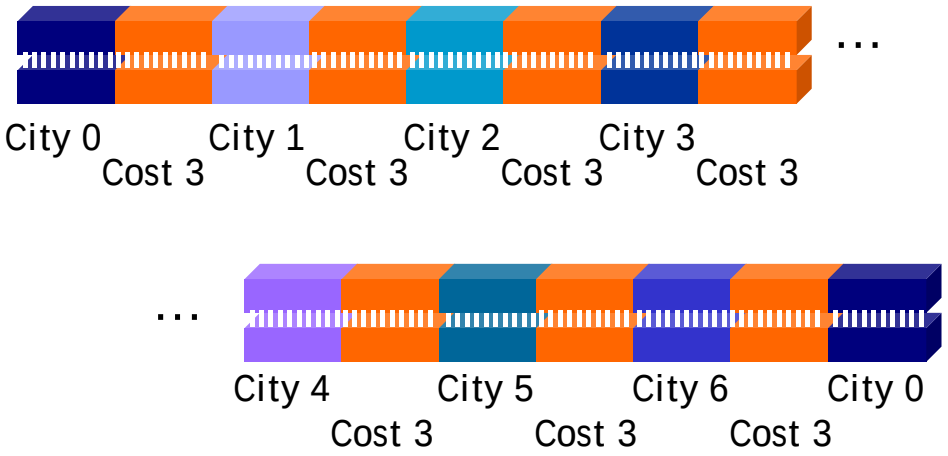
- TGGE
  (Temperature Gradient – Gel Electrophoresis)



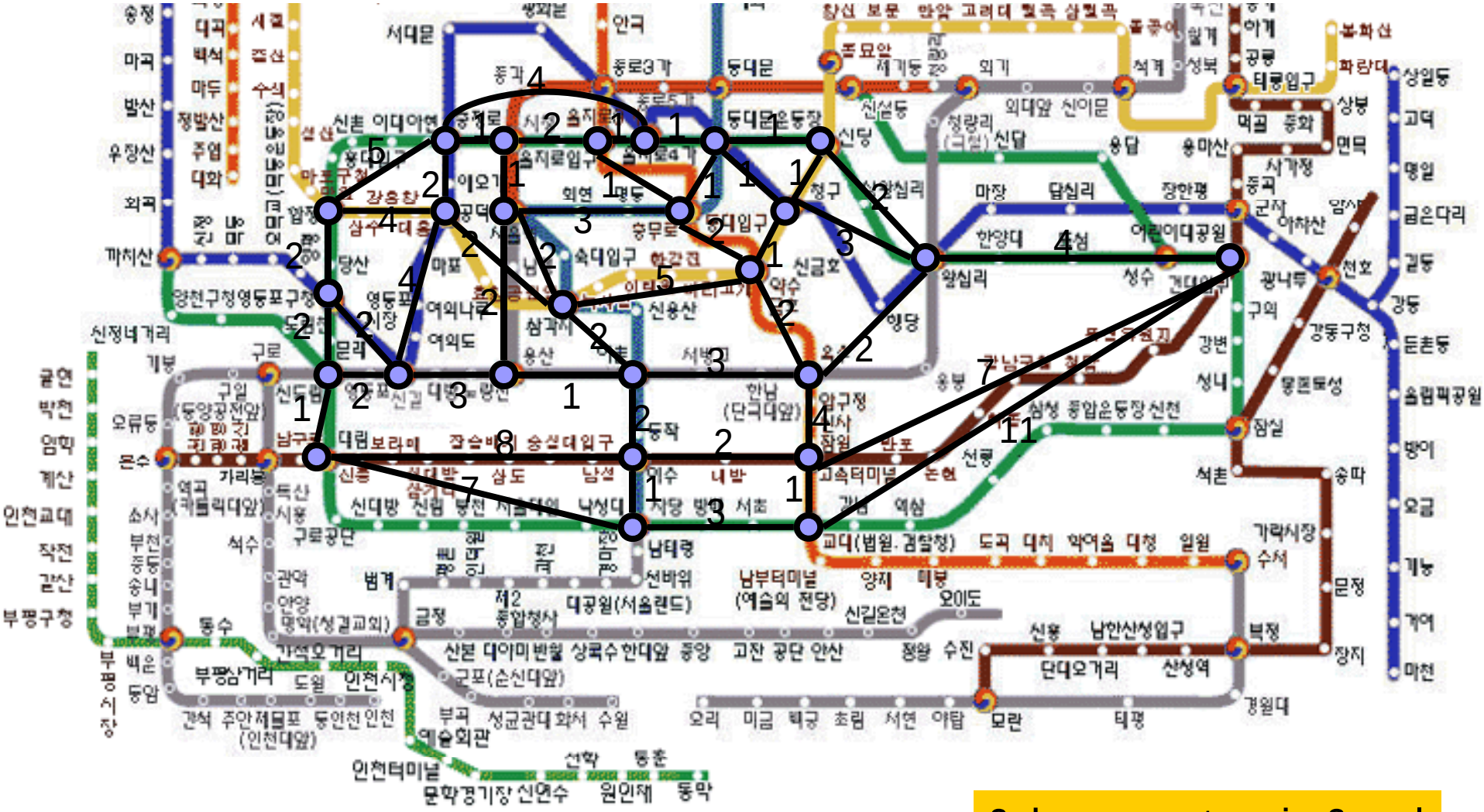
Readout

(by cloning and sequencing)

……TTCTGCGTTGTTTCGGGGTACAGTGGCTCCTCCGTT
CCGCCTGCACTGTGGAGAGGTGAGCAGTGGCTCCTCCGTT
CCGCGTGGATTCACAAGGCCATCGCAGTGGCTCCTCCGTT
CCGCATACGGCGTGGTTTTTCGGGCAGTGGCTCCTCCGTT
CCGCAAACGGTCGTAAGTGATGAACAGTGGCTCCTCCGTT
CCGCGCACAGTCCACCTGTAGACACAGTGGCTCCTCCGTT
CCGCTATGTCCAGCTGTCGCAAAGCAGTGGCTCCTCCGTT
CCGCTTCTGCGTTGTTTCGGGGTA……

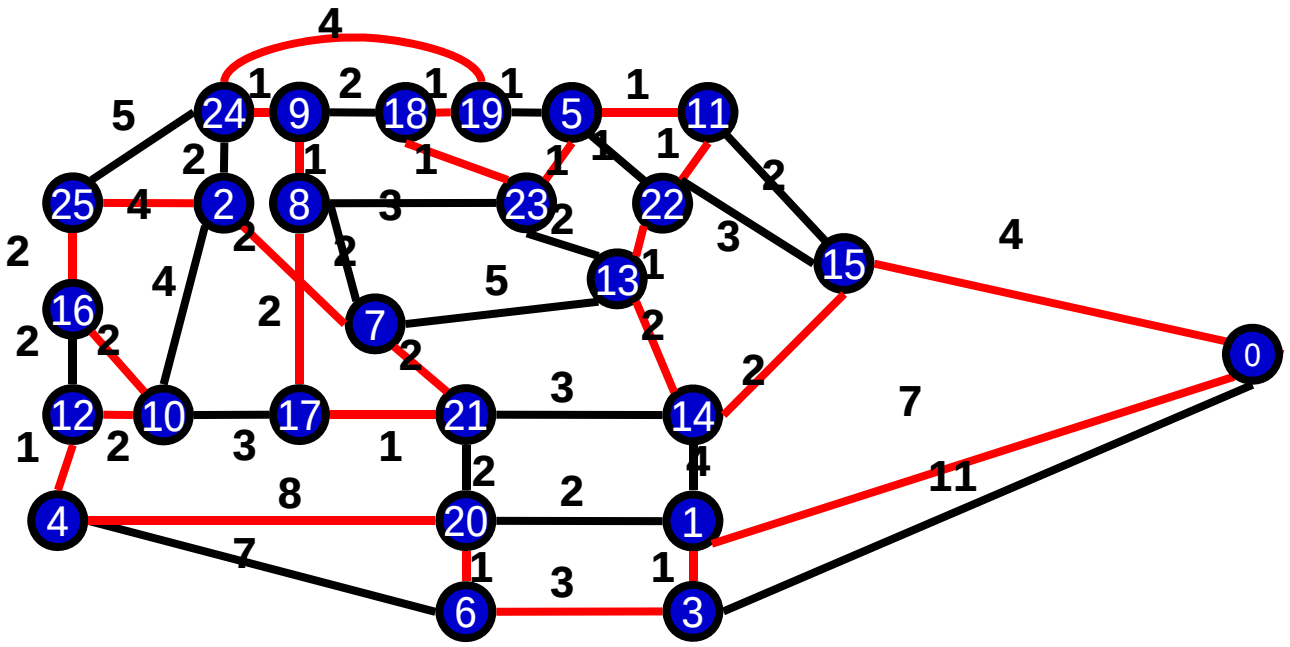


Toward Larger Problems



Subway routes in Seoul

Target Problem: 26-City TSP



- Graph with 26 vertexes (cities) and 92 edges (roads)
  - **Vertex**: station connected with more than two stations
  - **Weight**: number of stations between vertex stations