Lecture 1.2

Introduction:
Strings that encode Life

An Historical Perspective

- … – 1900 Pre-Mendelian period
- 1900 – 1940 Pre-DNA period
- 1940 – 1990 DNA period

►1990 – 2003 Genomic period

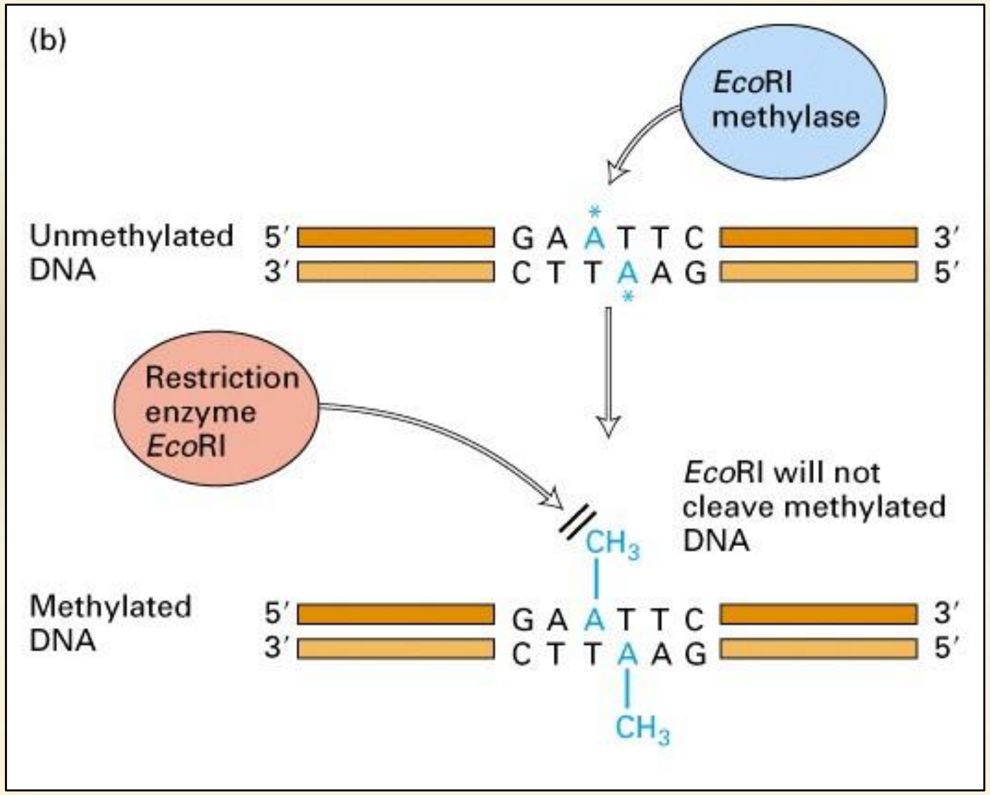
- 2003 – … Post-genomic era

Modern Biology

- Mechanism
- Cell theory
- Evolution

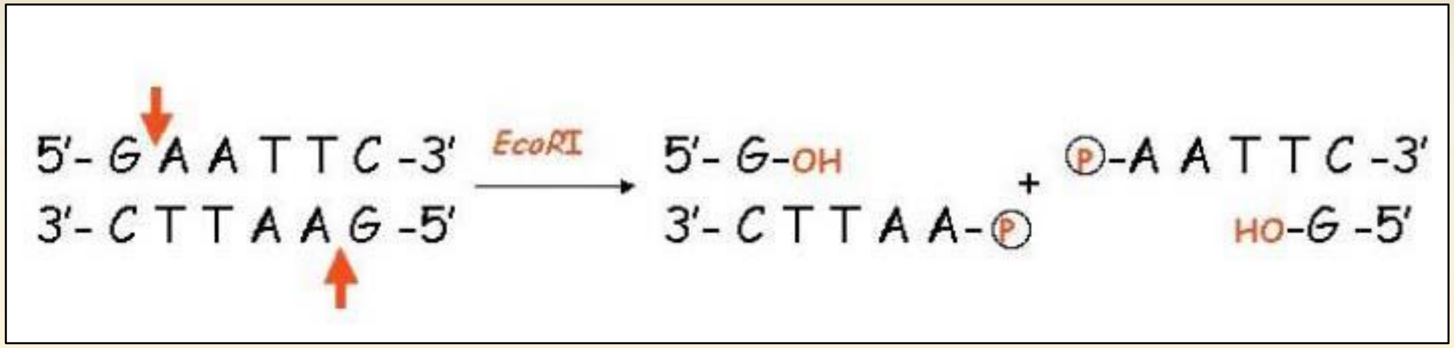
Technology: Manipulating DNA

- Restriction enzymes

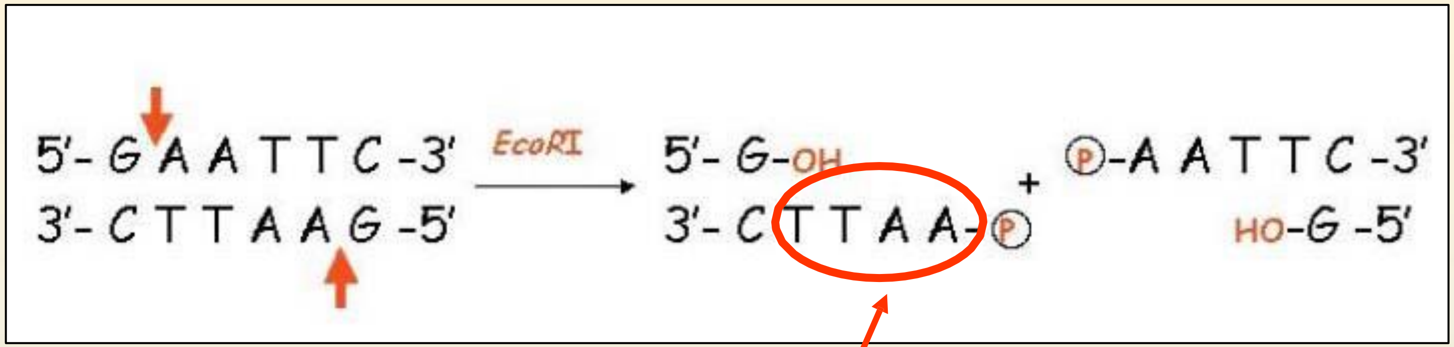


Technology: Manipulating DNA

- Restriction enzymes
  - Can cut DNA duplex at specific sites (palindrome sequence).
  - Do not discriminate between DNA from different organisms
  - A natural part of the bacterial defense system
  - High specificity for their recognition site means that DNA will be cut reproducibly into defined fragments



Technology: Manipulating DNA



- Restriction enzymes
  - Produce *sticky ends* of a single-stranded DNA which can base-pair (anneal) with any complementary single-stranded DNA sequence

Technology: Manipulating DNA

- Restriction enzymes
- Cloning vectors – replicating systems in addition to chromosomes:
  - Plasmids and *BACs* in Prokaryotes
  - Artificial chromosomes in Yeasts (Eukaryotes), *YACs*
  - Detailed restriction map of cloning vector
  - Marker – antibiotic resistance

Technology: Manipulating DNA

- Restriction enzymes
- Cloning vectors
- Reverse transcriptase
  - makes transcription from RNA to DNA (retroviruses – HIV)
  - we can take a mRNA (unstable) of any expressed gene and transcribe it into the DNA sequence (stable, double-stranded)
  - this DNA is called *cDNA*

Technology: Manipulating DNA

- Restriction enzymes
- Cloning vectors
- Reverse transcriptase
- Recombinant DNA
  - Self-replicating system containing artificially introduced gene
  - Example: production of insulin
  - Future: production of spider silk, biodegradation of waste

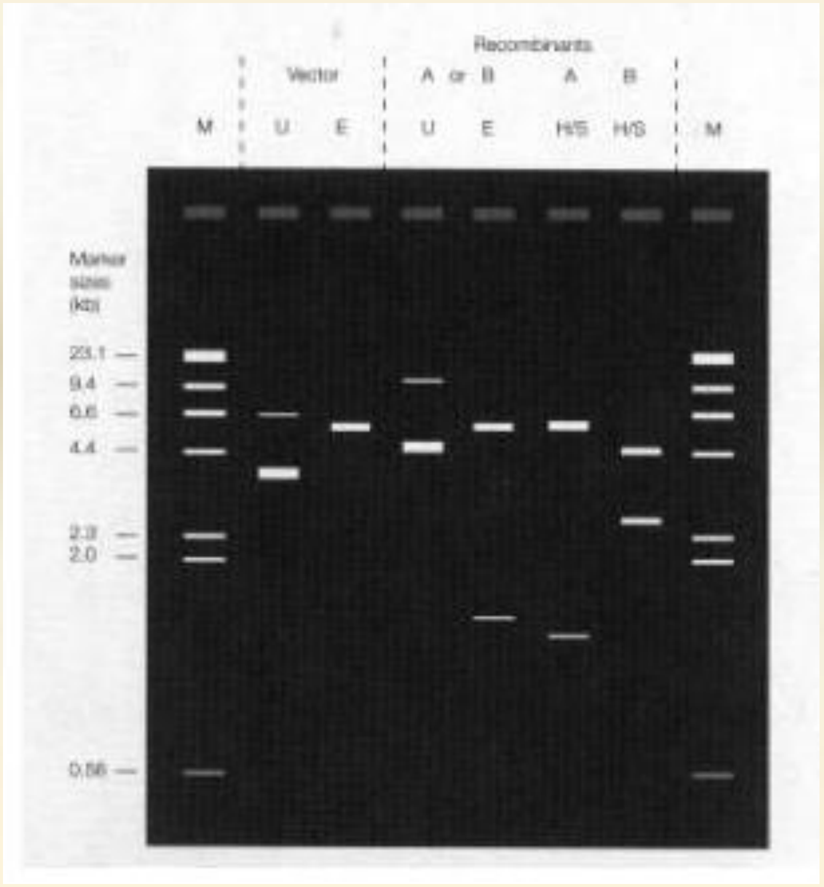
Technology: cDNA libraries

- Produce cDNA of a gene
- Clone this DNA in BAC, YAC or plasmid
- The amount of DNA sequence can be increased using Polymerase Chain Reaction (PCR)

Technology: electrophoresis

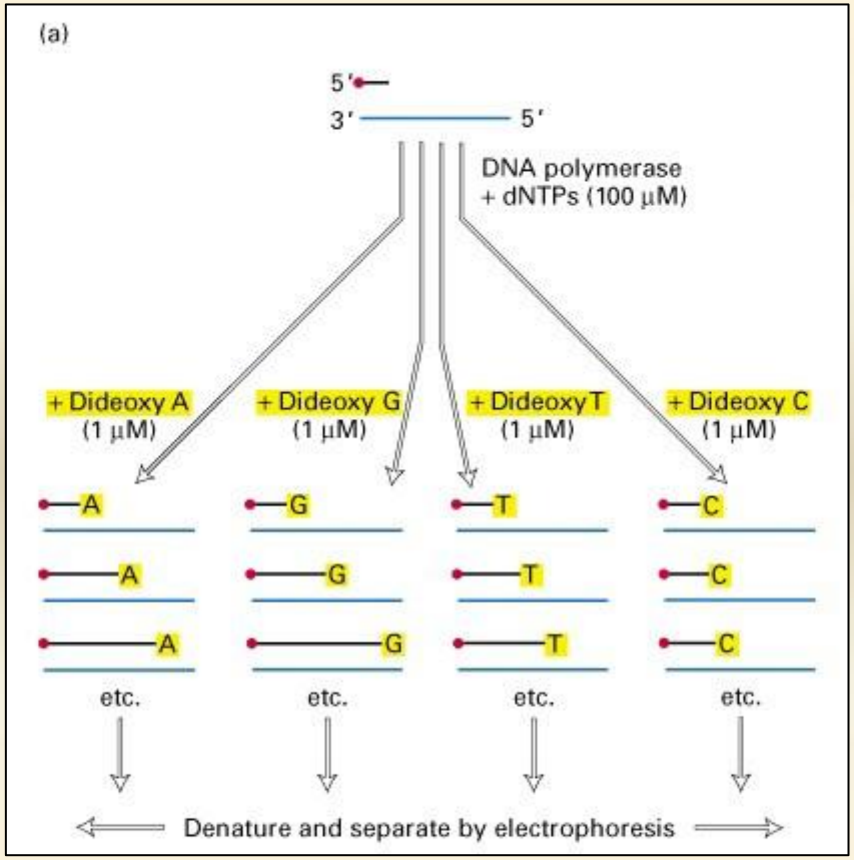
- Gel electrophoresis – determine length of DNA fragments

The length of DNA molecules is *decreasing* – smaller molecules run faster in a porous gel



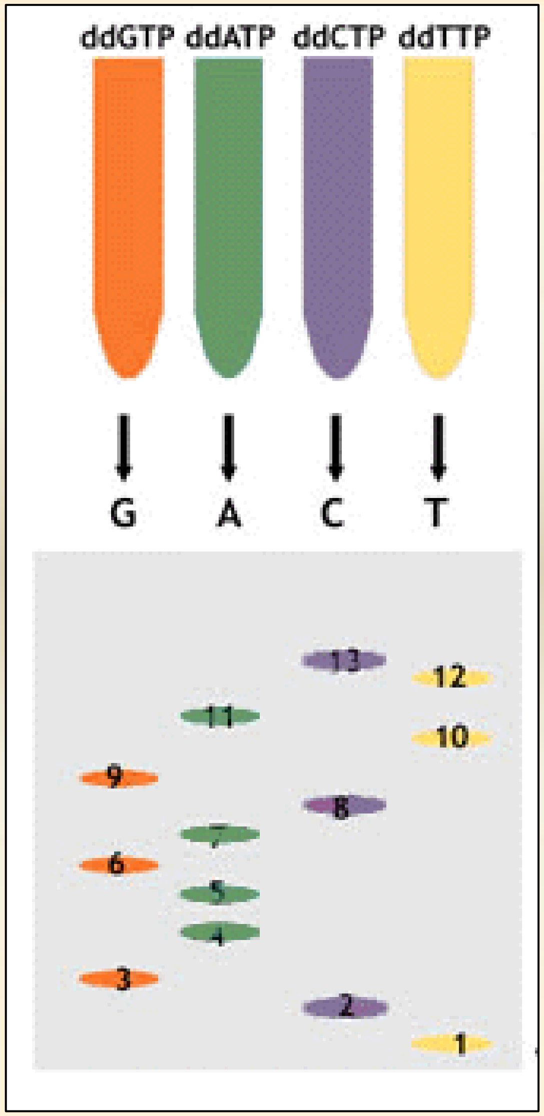
Sequencing

- Enzymic chain termination method
  - 4 different reaction tubes
  - Primer – sequence complementary to the start of the sequenced DNA
  - Mix of A,C,G,T radioactively labeled nucleotides
  - Small amount of dideoxynucleotides – when incorporated, no further chain growth



Sequencing

- The resulting DNAs from 4 tubes are loaded into 4 adjacent lines of the gel
- We can read the sequence from gel
- The sequence is read bottom up – from shorter to longer
- Process is automated
- Only up to 1000 nucleotides can be sequenced at a time
- We can determine sequence of all cDNAs in a cloning library



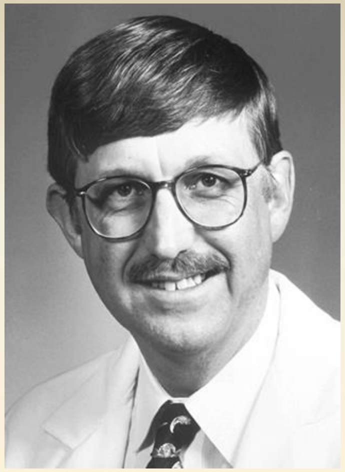
Sequencing genomes

- 1985 – proposal to sequence entire Human genome. Financed by US Department of Energy (DOE), lead by Watson, at first, then by Francis Collins
  - "The fear is not *big* science so much as *bad* science," said Botstein, “the DOE's proposal is a scheme for unemployed bombmakers."
- First, model organisms were sequenced
  - E. coli (bacteria)
  - Drosophila (fruit fly)
  - C. elegans (round worm)

Human Genome Project – 1986-2003

The scientific value seemed dubious. Although many biologists agreed that maps of the chromosomes would be useful for finding genes, what good would come from deciphering every A, T, G, and C, especially since most of them were "junk" that did not code for genes.

Controversial From the Start
Why sequence junk? → Human Genomes and Cancer research


Francis Collins

Craig Venter

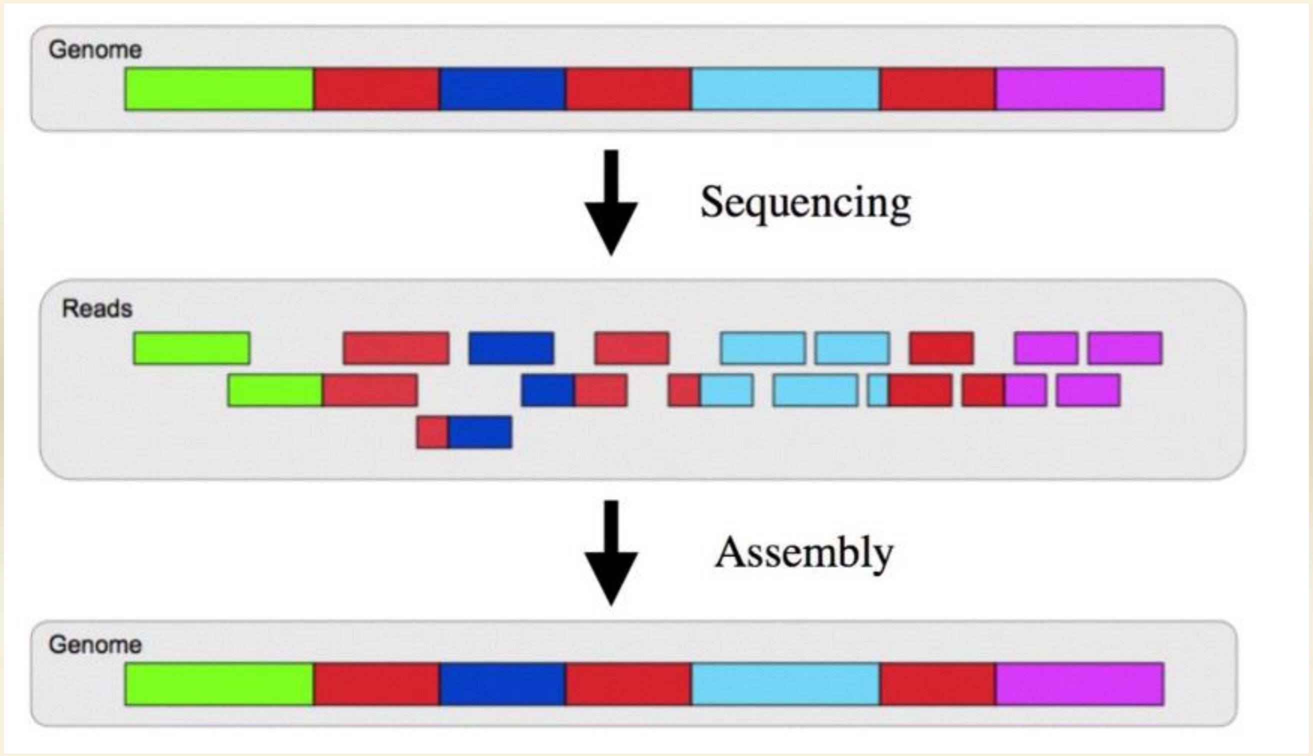
Human Genome Project

- 1985-the project initiated by Charles DeLisi, head of the department of energy (DoE) in the USA
- 1990-launched with the intention to be completed within 15 years and with a 3 billion dollar budget
- 1996-”Bermuda principles” – formalized the release of sequence data into public databases
- 1998-Craig Venter forms *Celera* company and promises to finish sequencing in 3 years with an ambitious “whole genome shotgun” approach
- 1999-the public project responds to Venter’s challenge and changes their target completion time
- December 1999-the first human chromosome sequence (22) published
- June 2000 – working draft announced
- February 2001 – the first draft published in Nature and Science magazines

The Human Genome Sequence

- $3*10^9$ basepairs (30 times larger than fruit fly and round worm – both around 10^8 basepairs), 250 times larger than Yeast genome
- Protein coding regions not more than 3%
- Around 46% of the remaining DNA – repeating sequences
- The rest contains promoters and other regulatory sequences

Genome Sequence Assembly



Genomic Assembly algorithms

- Greedy assemblers

  The assembler greedily joins together the reads that are most similar to each other.

- Overlap-layout consensus

  The relationships between the reads can be represented as a graph, where the nodes represent the reads and an edge connects two nodes if the corresponding reads overlap.

Genome Assembly problem: toy example

Find a string whose all substrings of length 3 are:

AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC.

All Substrings of Length 3

```
DISCRETE

DIS
 ISC
  SCR
   CRE
    RET
     ETE
```

All Substrings of Length 3

```
DISCRETE
DIS
 ISC
  SCR
   CRE
    RET
     ETE
```

Every two neighbor 3-substrings have
a common part of length 2, called an overlap

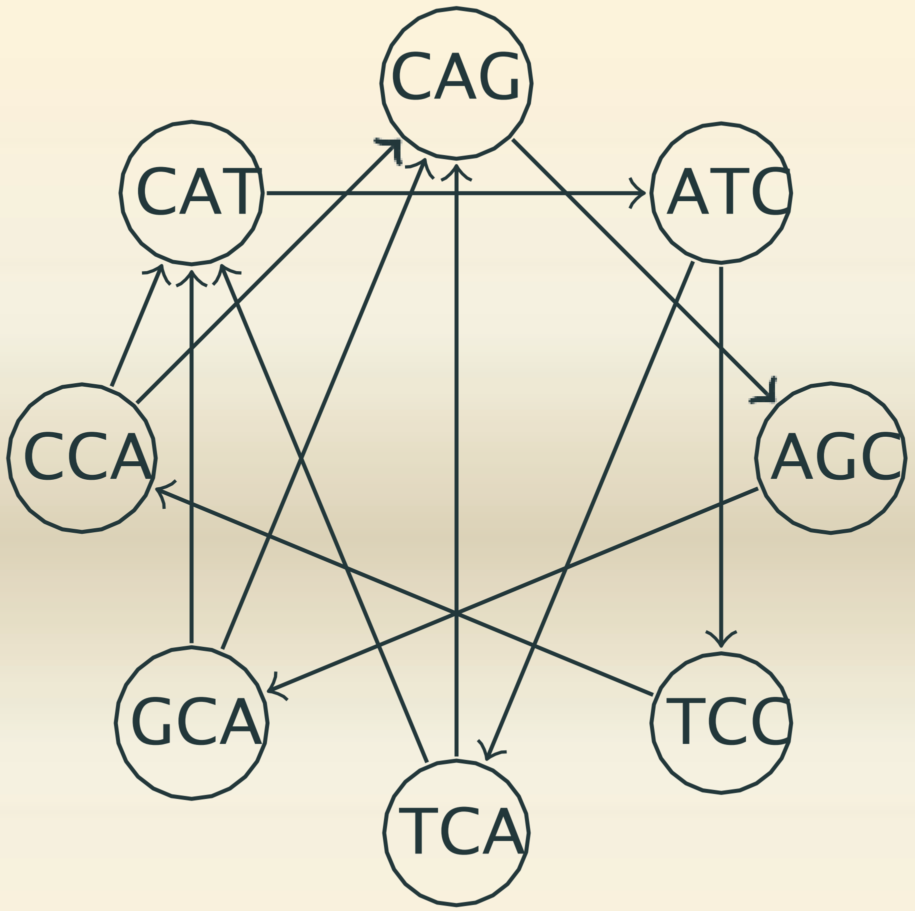
Computing a Permutation

- Algorithmic problem: Find a string whose all substrings of length 3 are AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC

- Hence, we need to order these 3-substrings such that the overlap between any two consecutive substrings is equal to 2

Overlap Graph

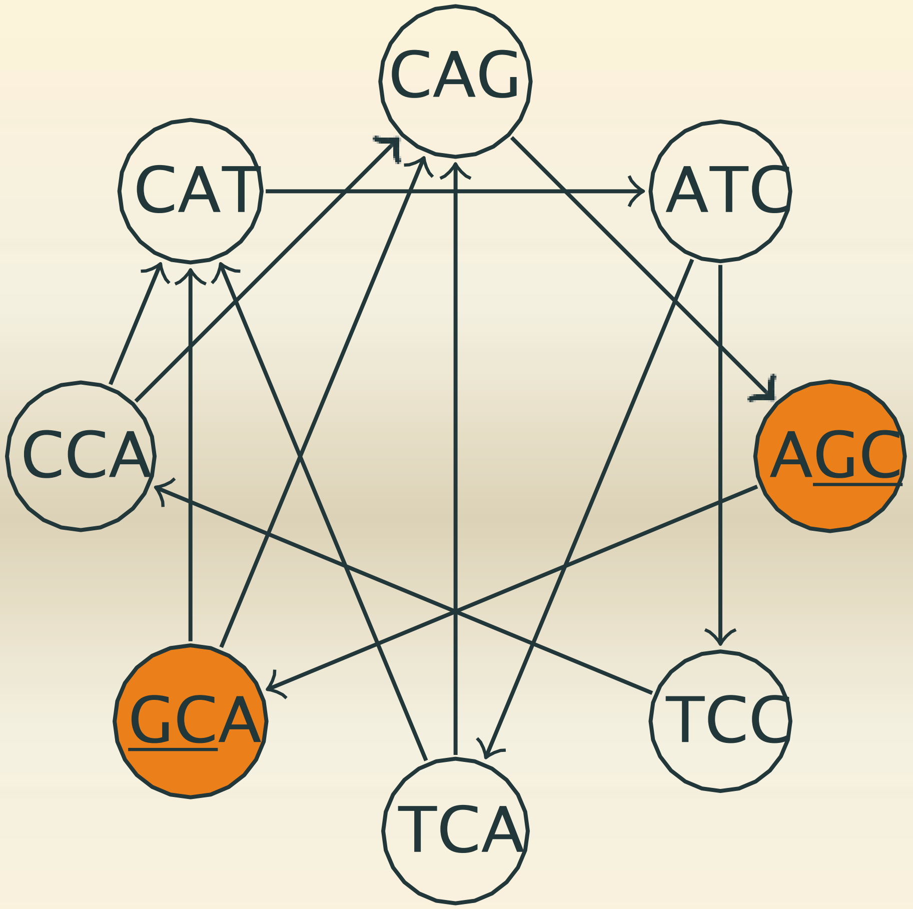
AGC
ATC
CAG
CAT
CCA
GCA
TCA
TCC



Nodes are substrings: short DNA sequence reads

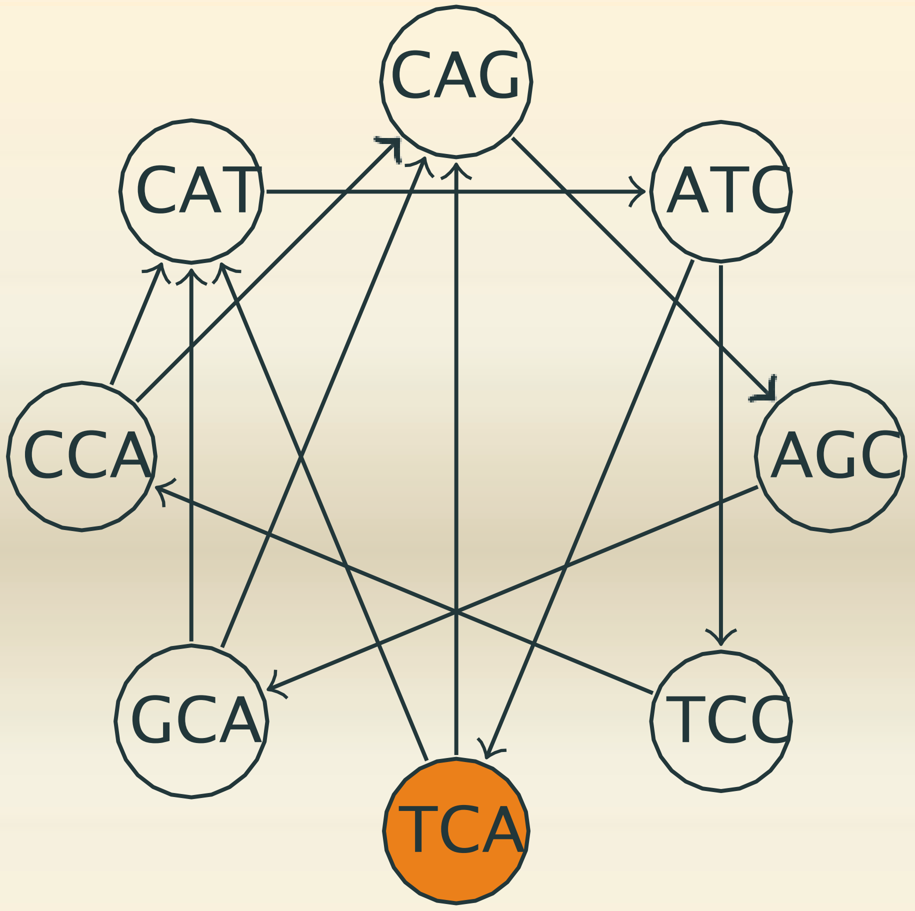
Overlap Graph

AGC
ATC
CAG
CAT
CCA
GCA
TCA
TCC



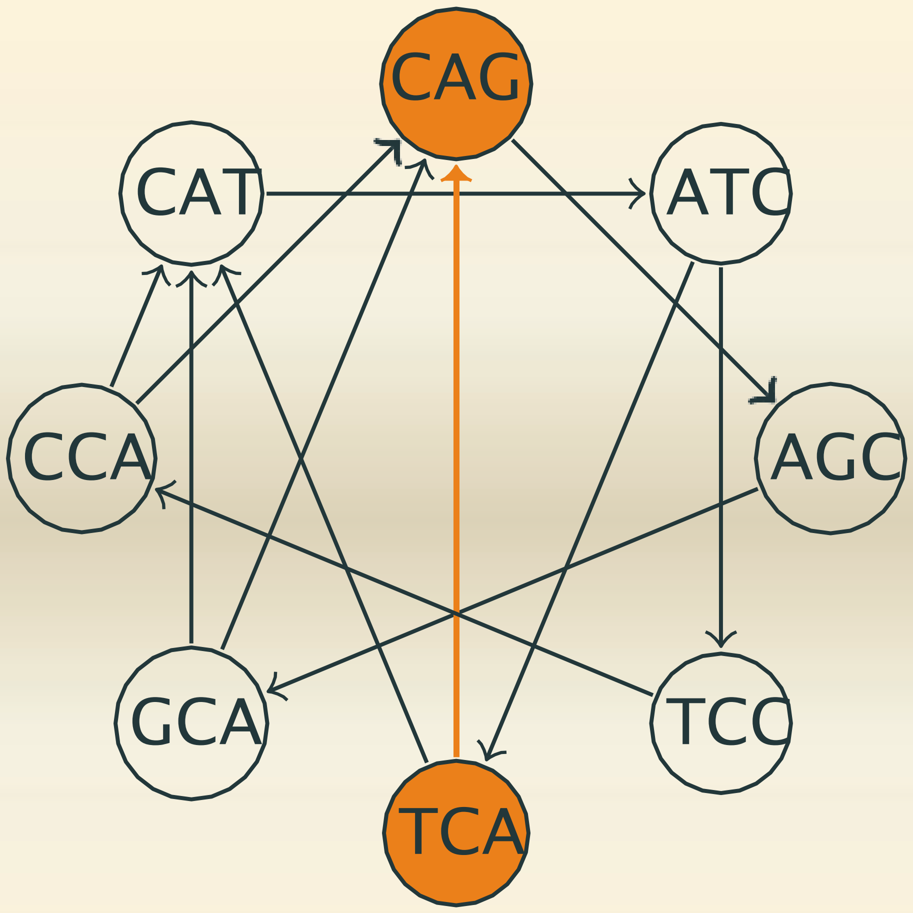
There is an edge from s_1 to s_2 if $s_1[2:3]=s_2[1:2]$

Hamiltonian path in the Overlap Graph



TCA

Hamiltonian path in the Overlap Graph



TCAG …

We solved Genome Assembly Problem!

- We modeled the problem of genome assembly as Hamiltonian path problem in the overlap graph!

We solved Genome Assembly Problem!

- We modeled the problem of genome assembly as Hamiltonian path problem in the overlap graph!
- But unfortunately we don’t have efficient algorithms for solving the Hamiltonian path problem!
- The approach is useless for the case when there are thousands or millions of input strings

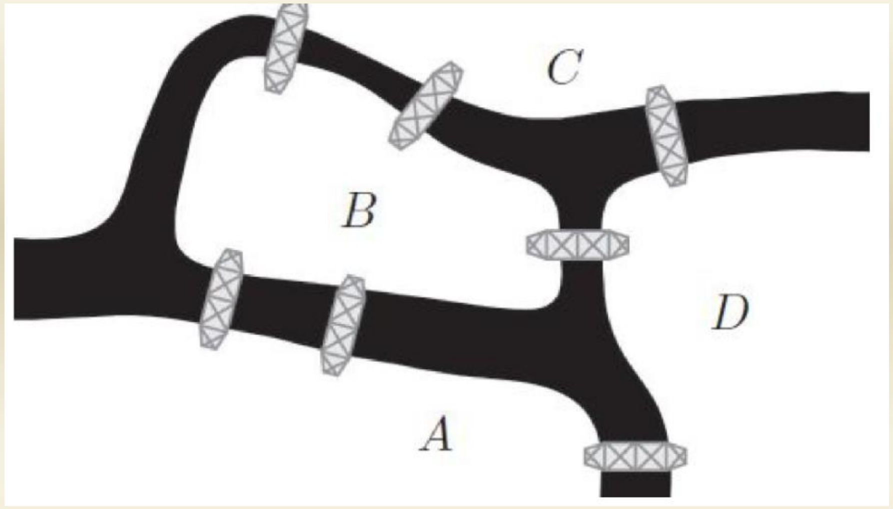
Computational biology

- The *bioinformatics* was born
  - the creation and advancement of databases, algorithms, computational and statistical techniques, and theory to solve formal and practical problems arising from the management and analysis of large-scale biological sequences.

Recall: Eulerian path problem

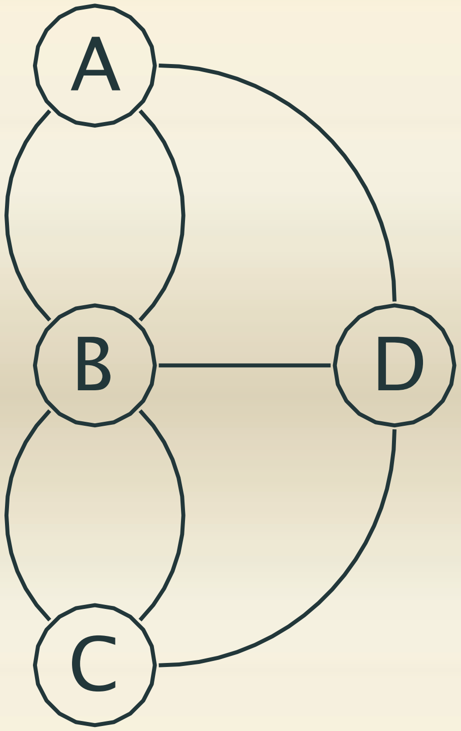
Leonhard Euler
1707 - 1783

Is there a path which visits **every edge** of the graph exactly once?



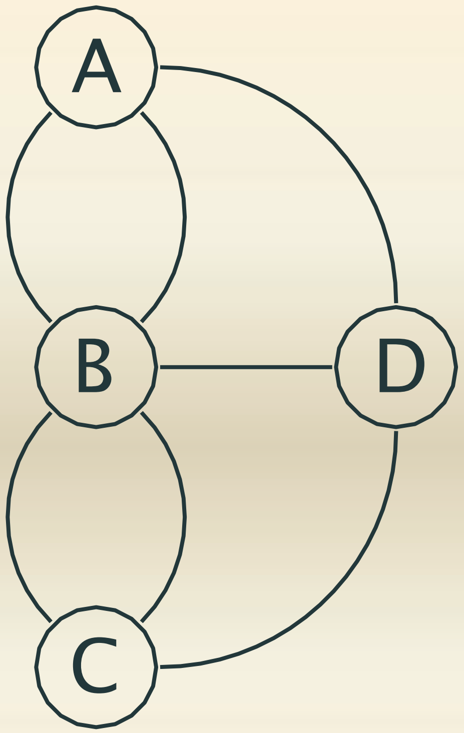
Seven bridges of Königsberg

$$\longrightarrow$$

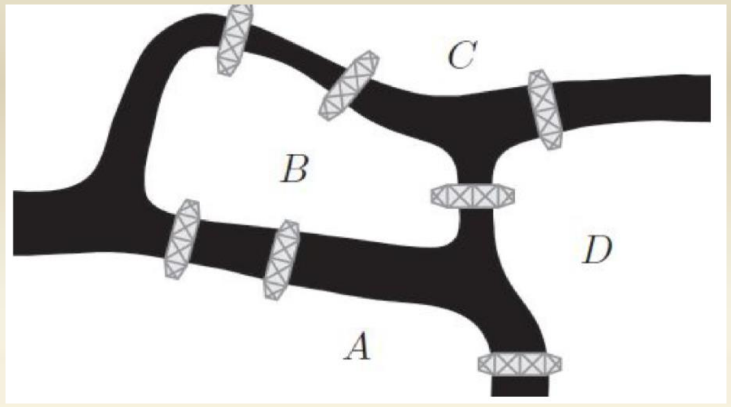


Modeled as Graph

Seven bridges of Königsberg



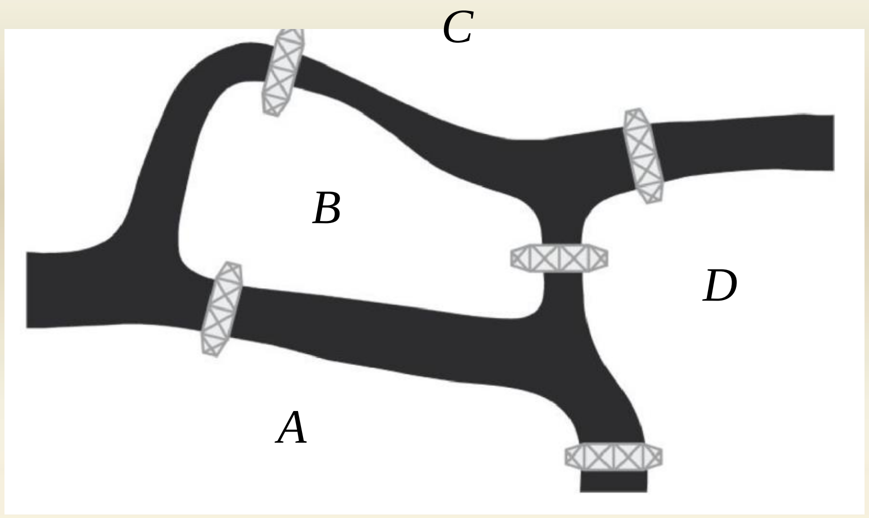
Is there an Eulerian Path through these seven bridges?



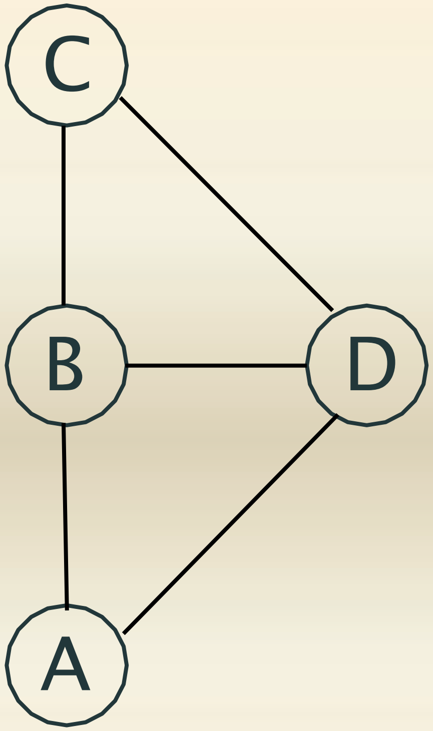
Königsberg, 17-th century

Five Bridges of Kaliningrad

Is there an Eulerian Path through these five bridges?



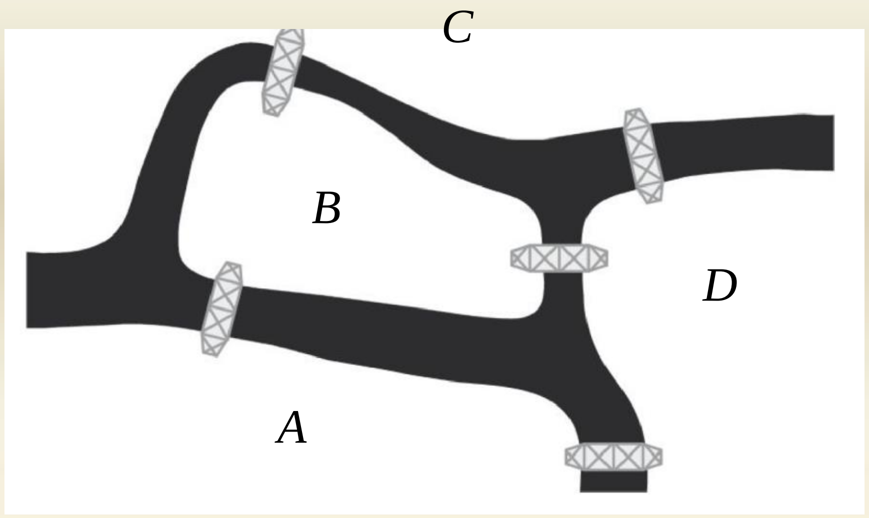
Königsberg (Kaliningrad), 21-th century



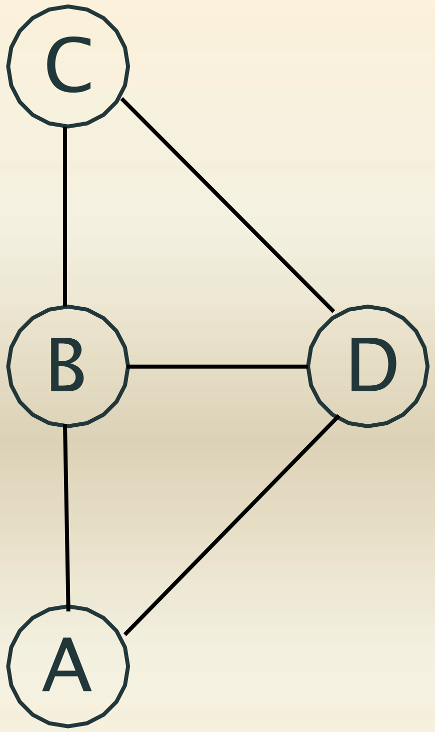
Five Bridges of Kaliningrad

B and D have odd degree

If there exists an Eulerian path, B and D must be START and FINISH

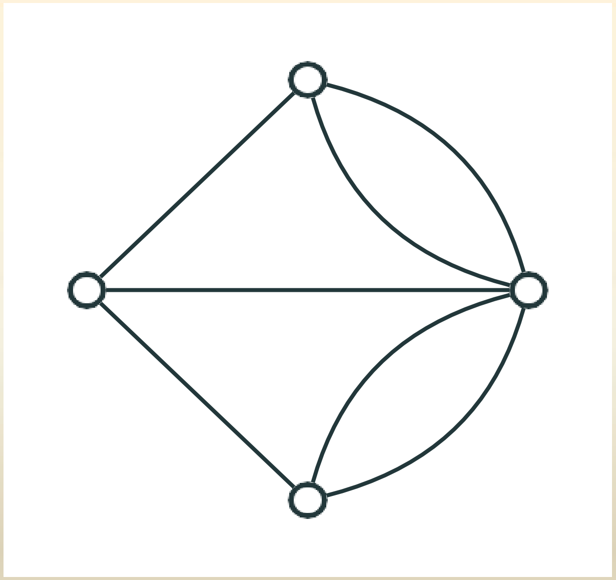


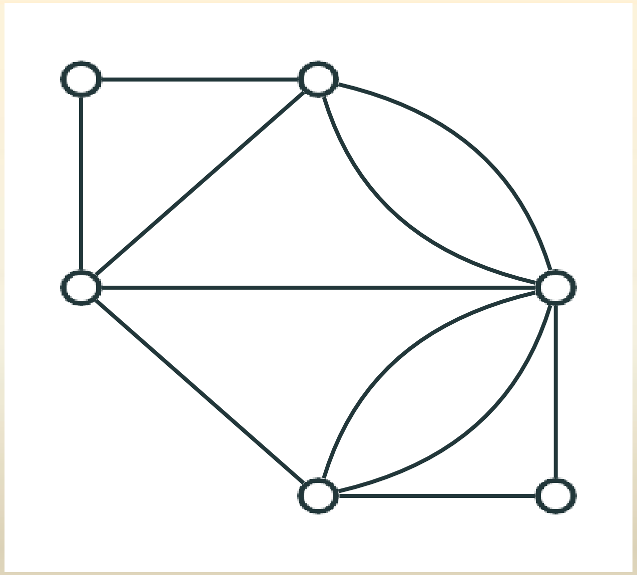
Königsberg (Kaliningrad), 21-th century



Graph A

Graph B

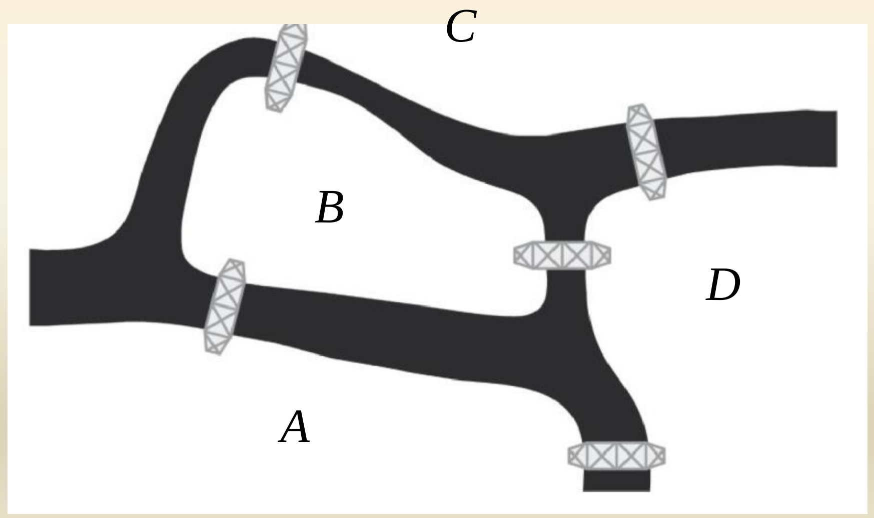


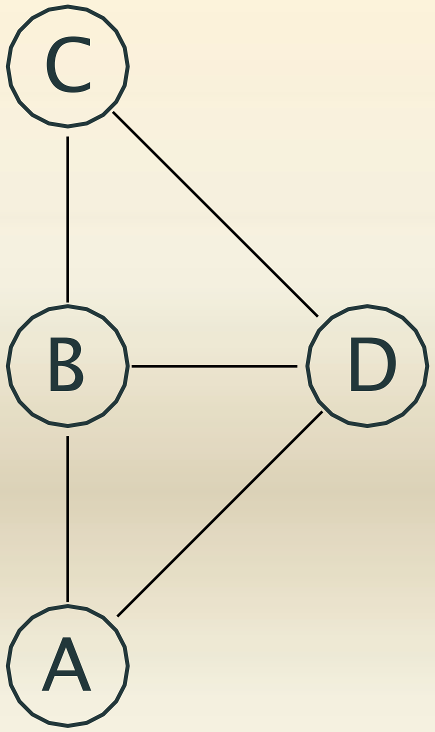


Which graph is an Eulerian graph (contains Eulerian path)?

A. Graph A
B. Graph B
C. Both A and B
D. Neither A nor B

Algorithm for finding Eulerian Path





The theorem about the existence of an Eulerian path can be transformed into an efficient algorithm for constructing it

Eulerian Path Algorithm

If there are no odd-degree vertices, start anywhere

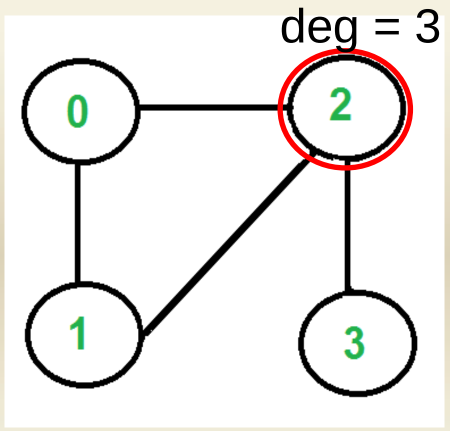
If there are 2 odd-degree vertices, start at one of them.

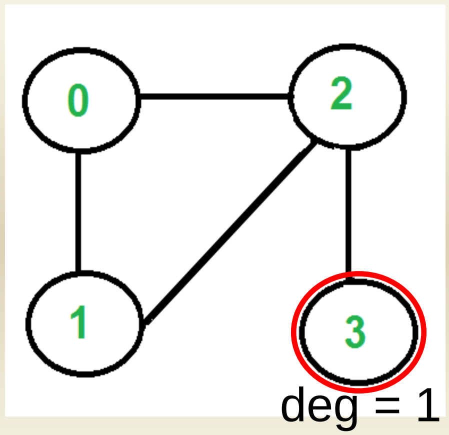
Out of the current vertex follow any edge

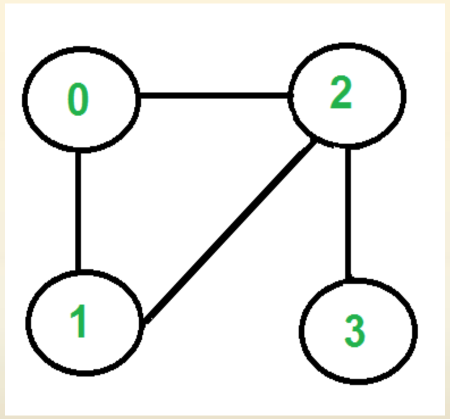
- If you have a choice between a *bridge* and a *non-bridge*, always **choose the non-bridge**: “don’t burn bridges“ so that you can come back to a vertex and traverse remaining edges
- Remove each followed edge (or mark as processed)

Stop when you run out of edges

Example

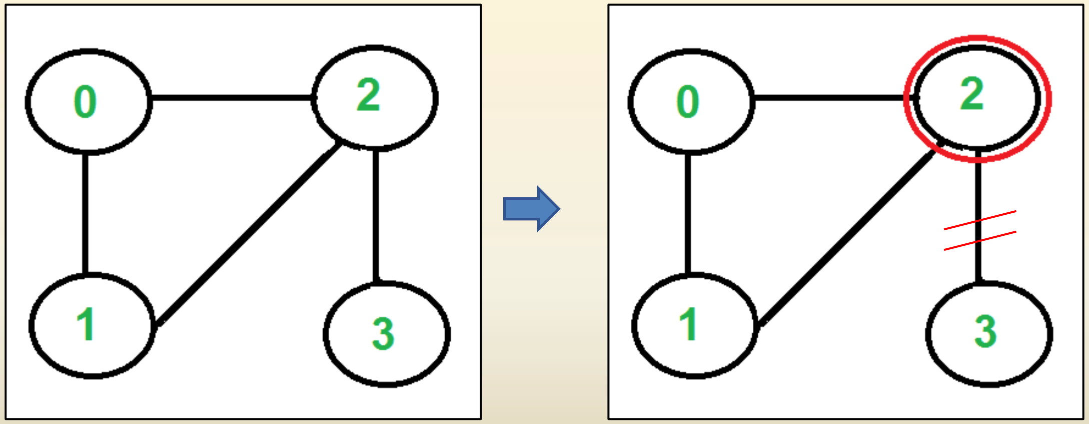






Two vertices with odd degree –
choose any of them to start

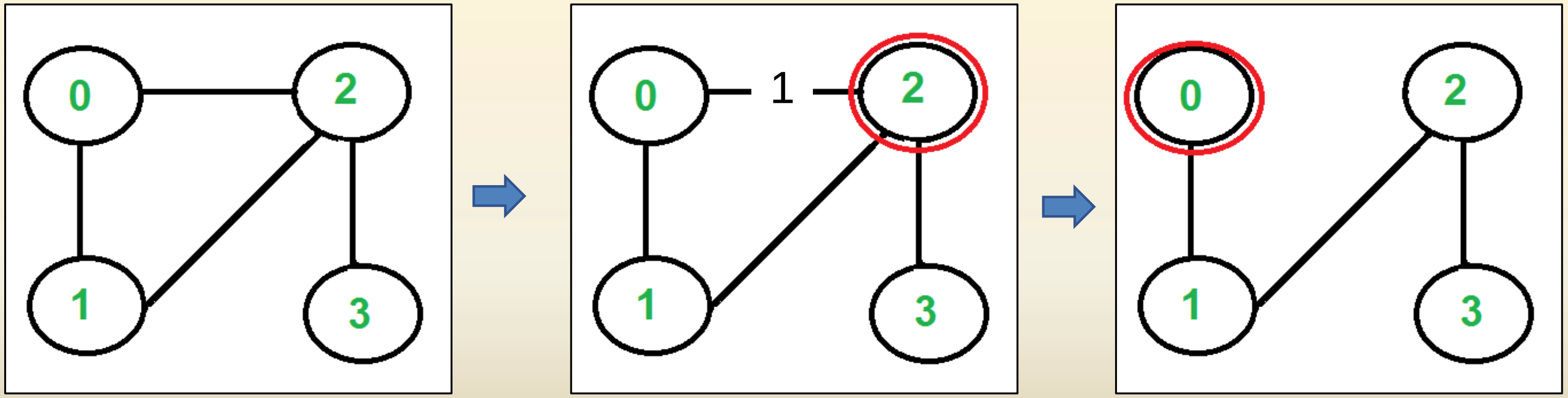
Example: where to go first?



Do not go there: (2,3) is a bridge

Eulerian Path:

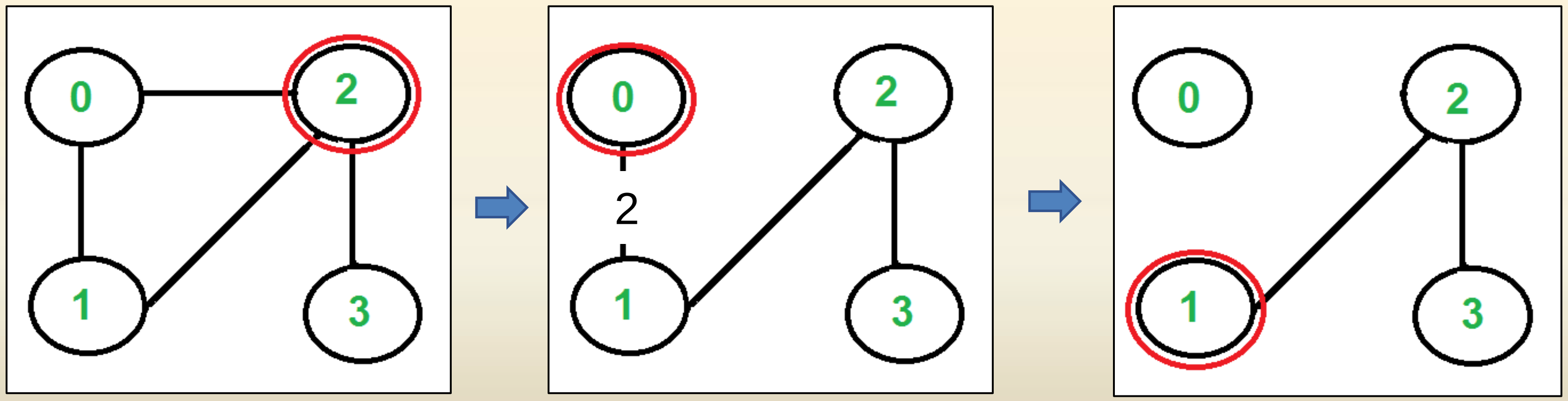
Example: step 1



Move along (2,0) and then delete edge (2,0)

Eulerian Path: (2,0)

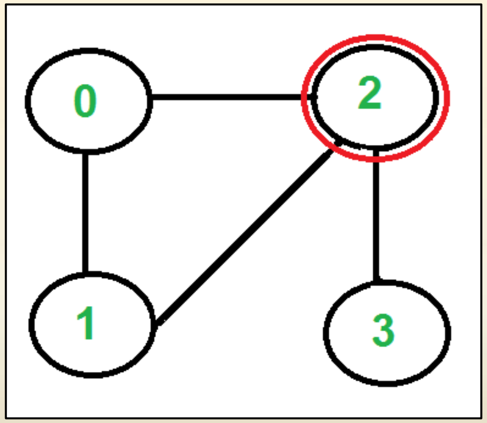
Example: step 2

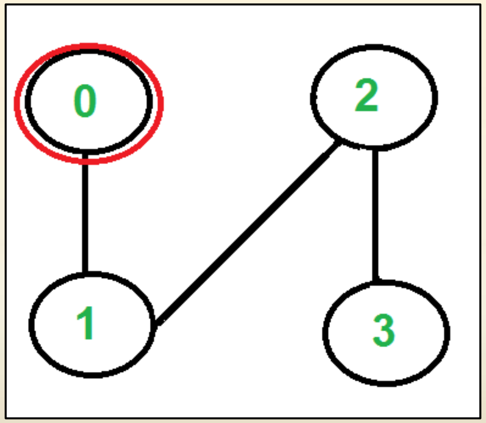


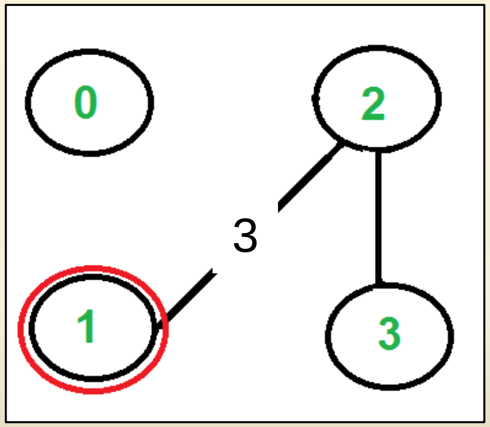
Move along (0,1) and then delete edge (0,1)

Eulerian Path: (2,0), (0,1)

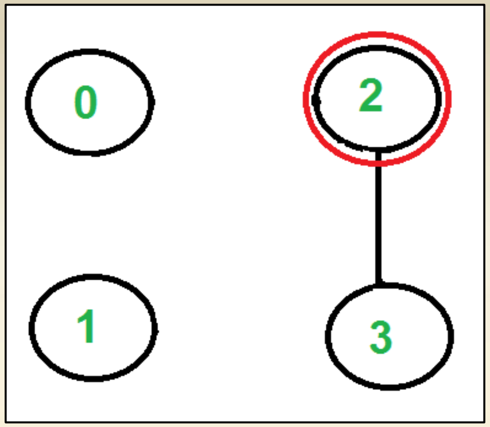
Example: step 3





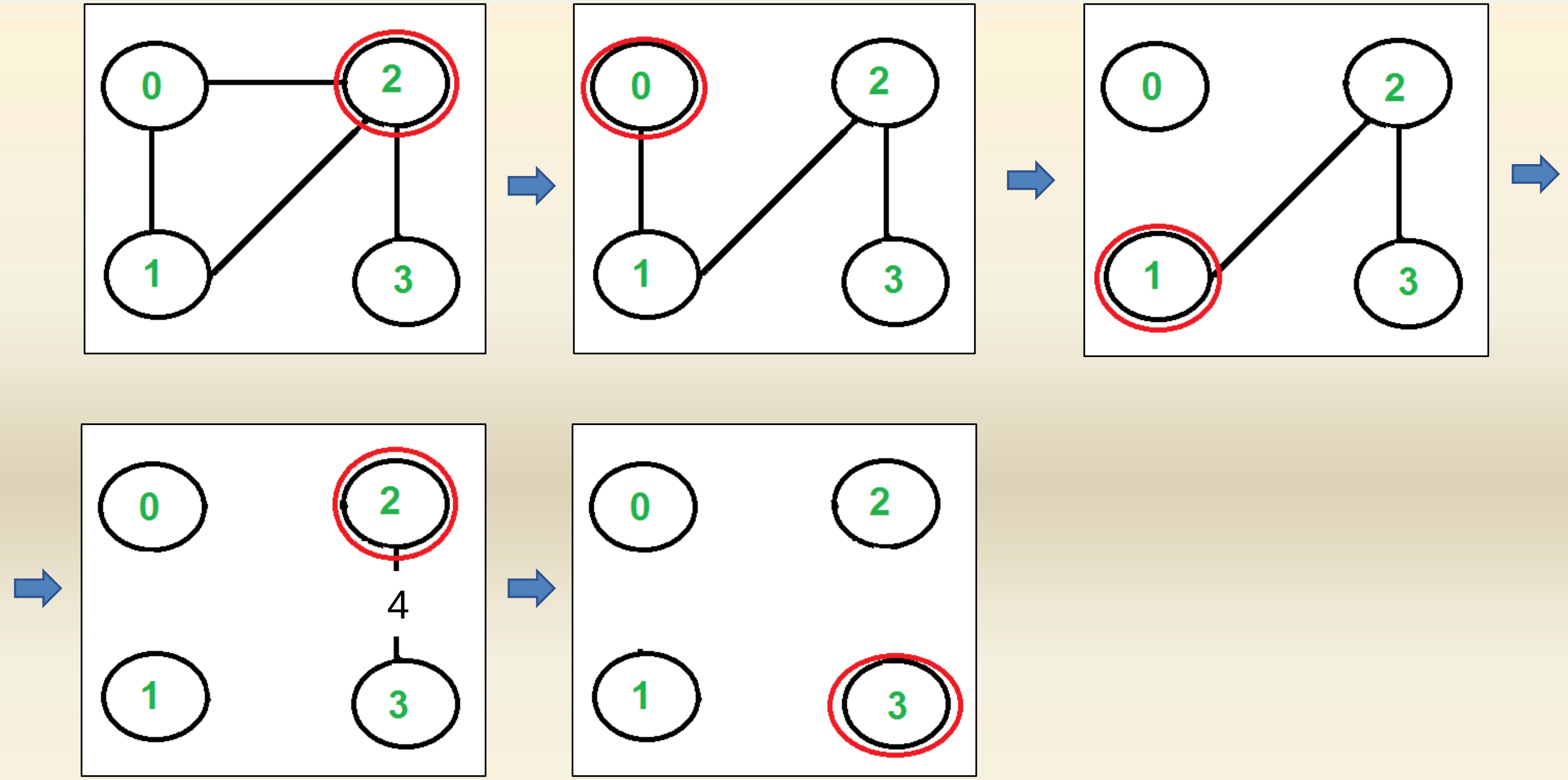


Move along (1,2)



Eulerian Path: (2,0), (0,1), (1,2)

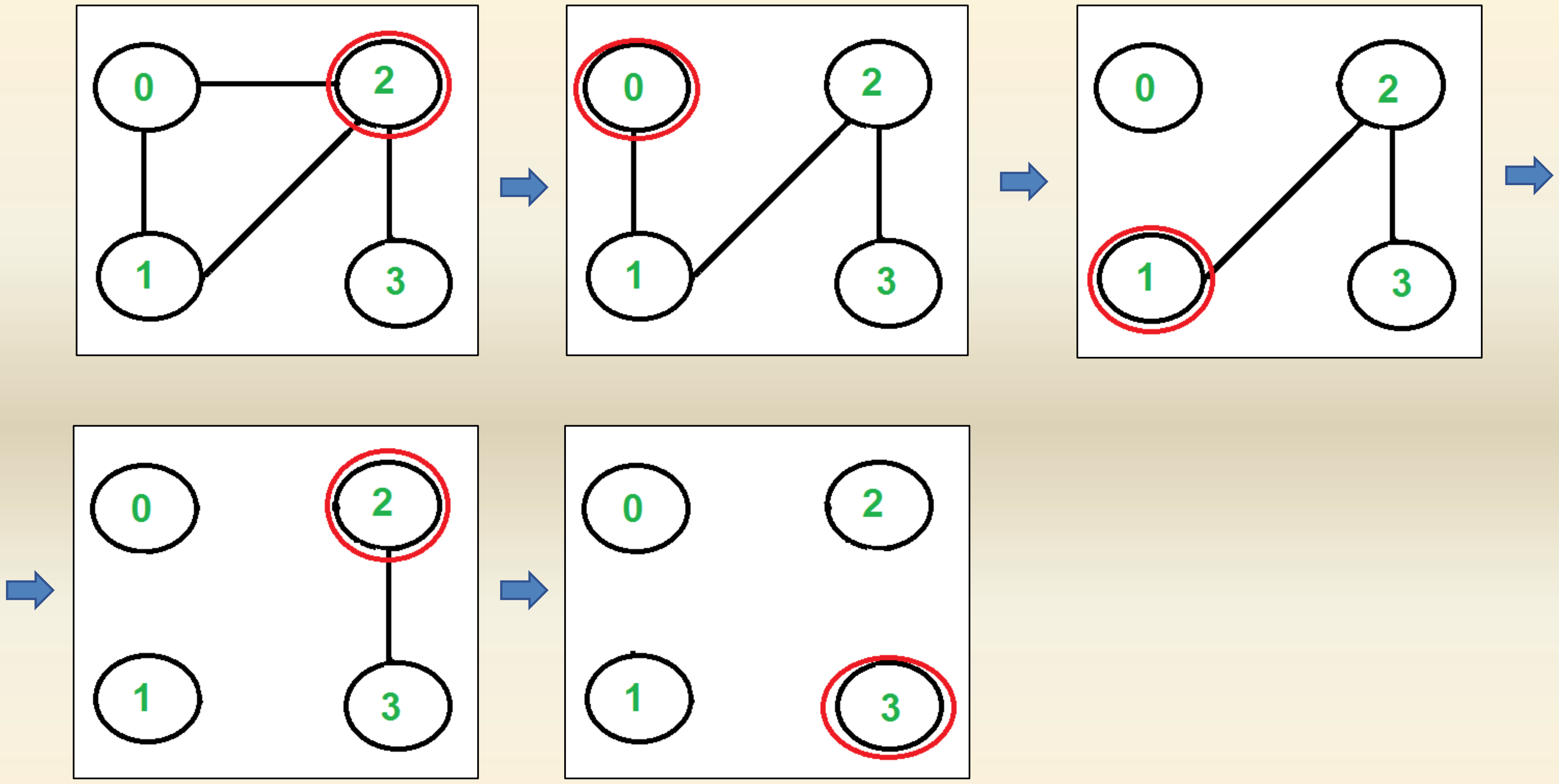
Example: step 4



Move along (2,3)

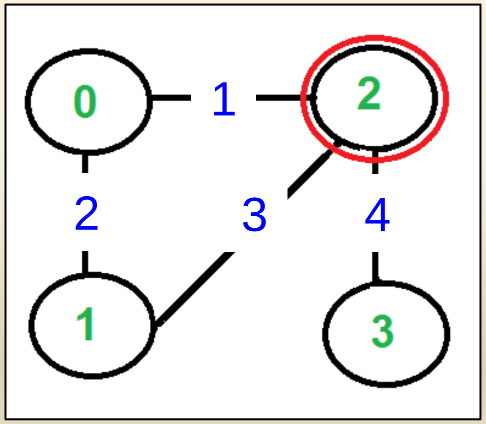
Eulerian Path: (2,0), (0,1), (1,2), (2,3)

Example: the end



Eulerian Path: (2,0), (0,1), (1,2), (2,3)

Example: the end



Eulerian Path: (2,0), (0,1), (1,2), (2,3)

Genome Assembly problem: still unsolved

Find a string whose all substrings of length 3 are:

AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC.

Different approach
(De Bruijn; Pevzner, Tang, Waterman)

State-of-the-art genome assemblers

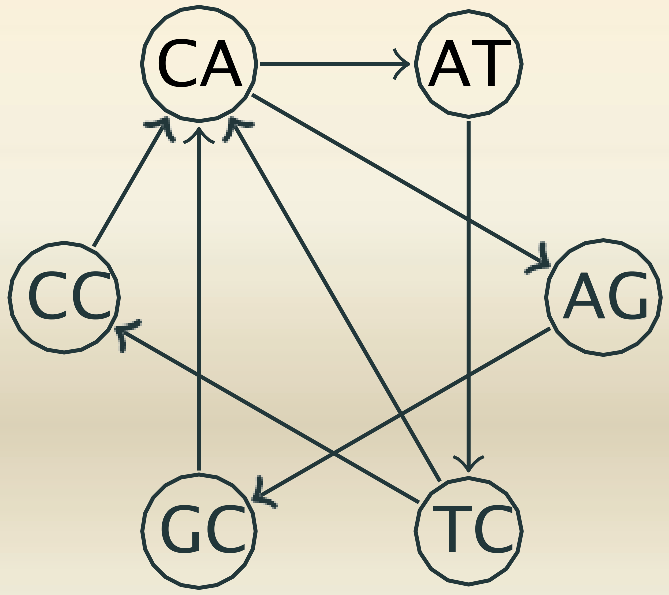
- In the overlap graph, each node corresponds to the input substring
- Let's instead represent each edge by the same substring, broken into 2 nodes (overlaps):

  E.g., represent the string CAT as an edge

  $CA \rightarrow AT$

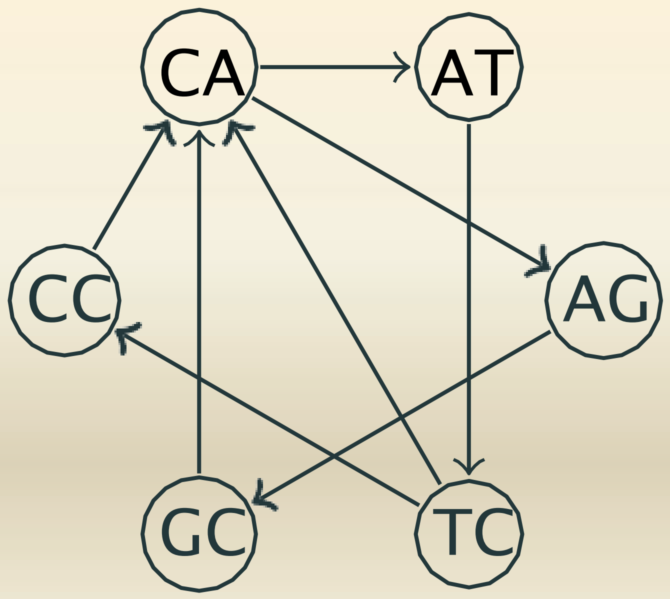
De Bruijn Graph

AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



De Bruijn Graph

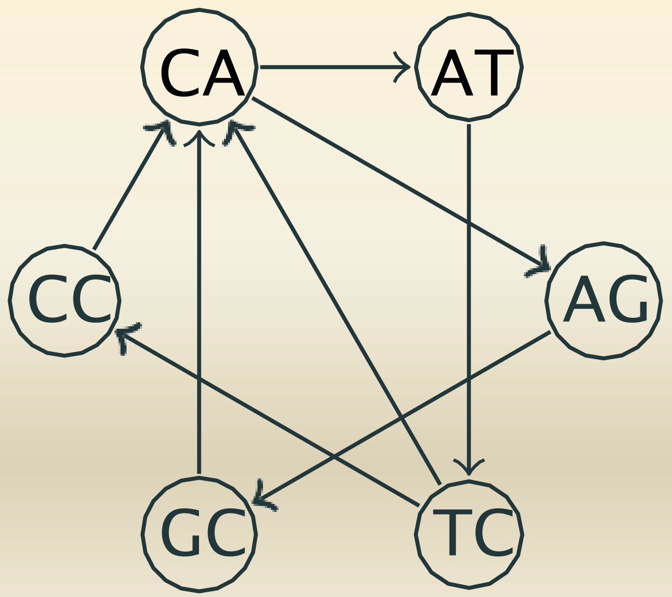
AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



now, we need to find an order of edges

De Bruijn Graph

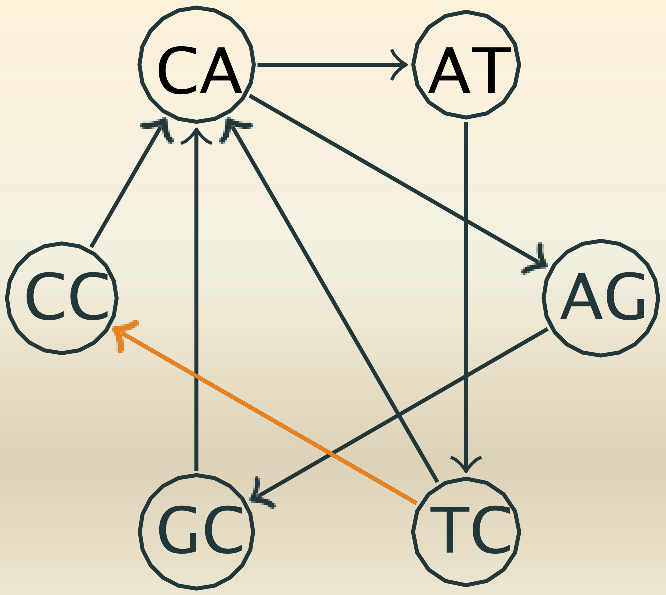
AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



that is, an Eulerian path

De Bruijn Graph

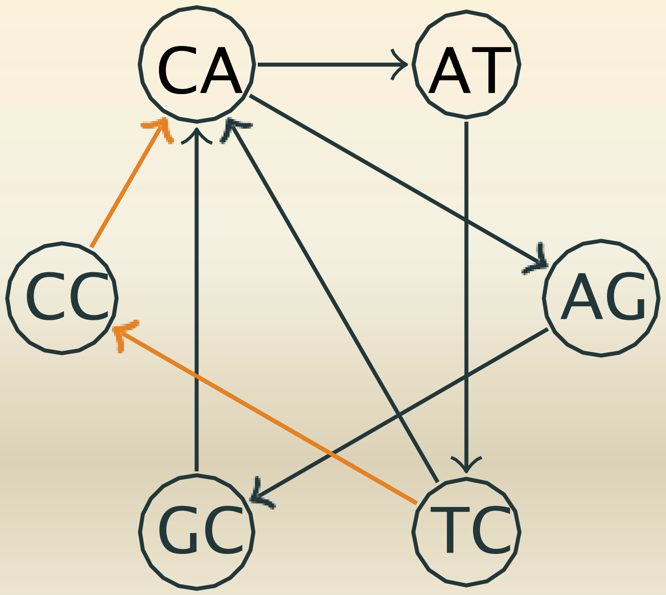
AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



TCC

De Bruijn Graph

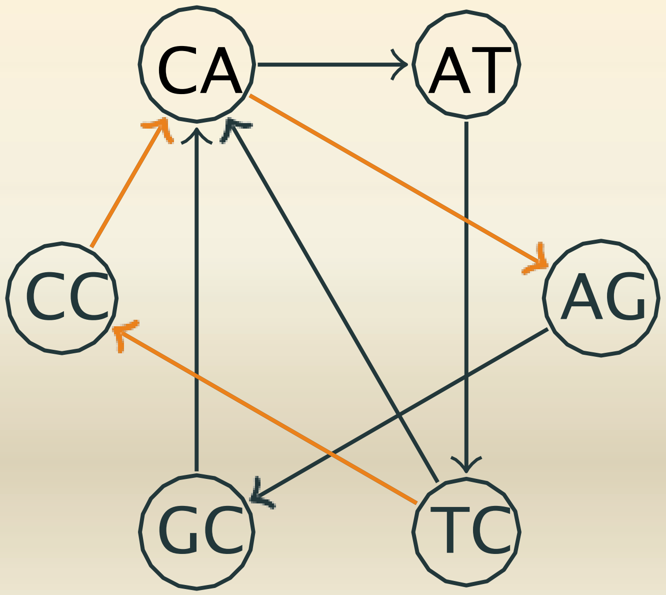
AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



TCCA

De Bruijn Graph

AGC, ATC, CAG, CAT, CCA, GCA, TCA, TCC



TCCAG

Activity: DeBruijn Graph

Imagine that you are given a **large** set of 3-letter strings which represent all possible different substrings of the large “genome” string:

him, eno, ome, chi, nom, mpg, pge, gen, imp

Recover the whole “genome” sequence by building a graph model of the problem.

Draw the graph and explain which algorithm you used on this model to recover the original “genome”