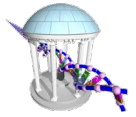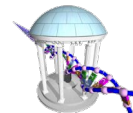# Comp 555 - BioAlgorithms - Spring 2021

- Genomes from short sequence fragments
- Using graphs to represent sequences
- Graph algorithms

THERE WILL BE A PYTHON CRASH COURSE TONIGHT FROM 5:00PM-6:30PM IN SN011.

Assembling a Genome

What we know about Genomes

- **DNA sequences are a biological system's hard drive**
  - They contain an operating system with all the low-level support for growing, dividing, and reproducing
  - They contain application programs for making cells that move our bodies, remember our mother's face, and store energy for use in lean times
  - They are robust. They have programs for repairing and replicating themselves. They even have backups!
- **DNA sequences vary in size**
  - Human nuclear DNA is composed of roughly 6 billion base-pairs distrbuted over 46 pairs of chromosomes
  - These 6 billion bases are comprised of 2 nearly identical copies
  - One of these copies is called a haplotype and its sequence is called a genome
  - Among humans, any two haplotypes are are 99.9% identical
- **How can we read off the sequence of DNA?**

DNA Sequencing History
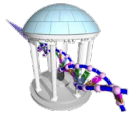
- **DNA sequencing was one of the most significant breakthroughs of the 20th century**
- **This was so inherently obvious it was awarded a Noble prize only 3 years after its development**

Sanger method (1977):

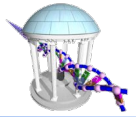Uses labeled dideoxynucleotide-triphosphates (ddNTPs) terminate DNA copying at random points.

Gilbert method (1977):

Used various chemicals (Dimethyl Sulfate, Hydrasine) to modify and then cleave DNA at specific points (G, G+A, T+C, C).
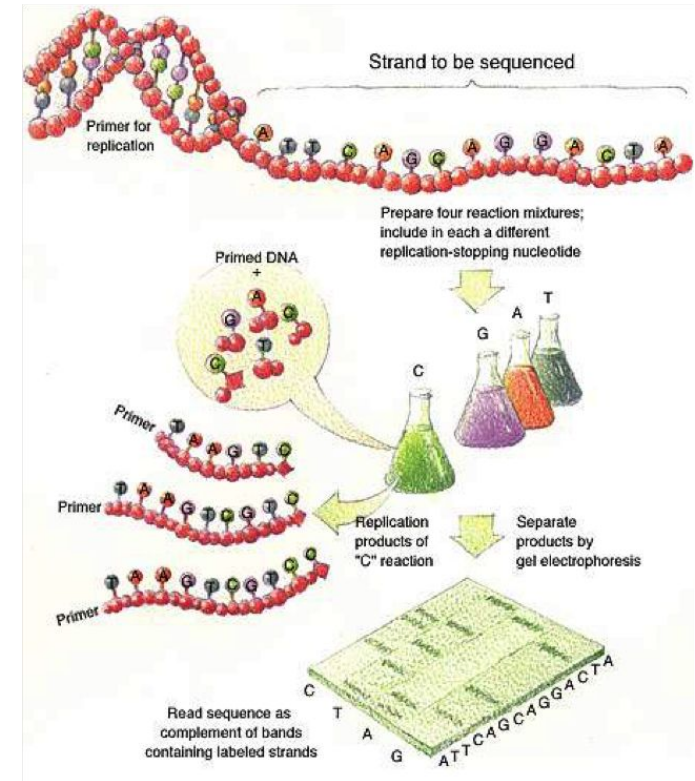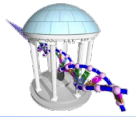
Fredrick Sanger

Walter Gilbert

Sanger Method

1. Use the polymerase chain reaction (PCR) to make billions of copies of a DNA sequence
2. Starting at *custom* primer, sort of like our the *origin of replication*, we initiate one last replication
3. Include *chemically altered* and *fluorescently labelled nucleotides*, called dideoxynucleotide-triphosphates (ddNTPs)
4. If a ddNTP gets incorporated into a sequence it stops further replication
5. Separate replication products by length, using gel electrophoresis
6. Good for 500-1000 bases, then the error rates grow and extension rate slows
7. About 10 bases-per-second or 9.5 years to read an entire genome if we could do it from beginning to end

Sanger Method

1. Use the polymerase chain reaction (PCR) to make billions of copies of a DNA sequence
2. Starting at *custom* primer, sort of like our the *origin of replication*, we initiate one last replication
3. Include *chemically altered* and *fluorescently labelled nucleotides*, called dideoxynucleotide-triphosphates (ddNTPs)
4. If a ddNTP gets incorporated into a sequence it stops further replication
5. Separate replication products by length, using gel electrophoresis
6. Good for 500-1000 bases, then the error rates grow and extension rate slows
7. About 10 bases-per-second or 9.5 years to read an entire genome if we could do it from beginning to end
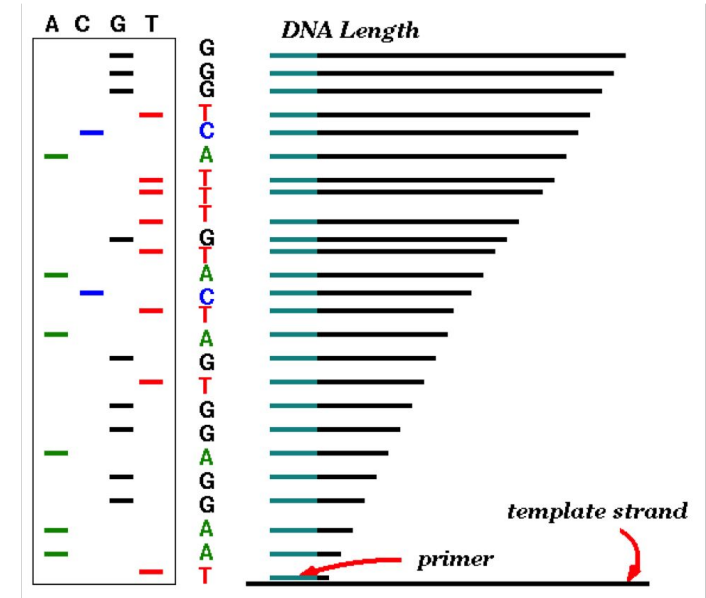
Assembling the Human Genome
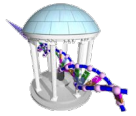
In 1990, a moon-shot-like project was begun to sequence the entire Human Genome.

- It would require 30x coverage to provide enough sequences
- Recall there are sequence differences-- Approximately 1:1000 bases
- Redundancy was needed to find the majority base from 16 different individuals (32 genomes)
- Also needed the extra coverage to assure that there is enough overlap to assemble the 500 base-pair reads

A $3 billion dollar NIH funded public effort led by Francis Collins with a 15-year plan. It would distribute the work across several labs in a community effort by assigning primers to groups on a first-come basis. New sequencing results yielded new primers, so the project required a central coordination.

In 1997 a private company, Celera, lead by Craig Venter, suggested they could beat the public effort by dispensing with primers. They'd just randomly fragment DNA and sequence each with no idea of the how sequenced fragments would fit together. In other words, they were going to rely on computer science to assemble their reads algorithmically.

The result was that, despite tensions, the groups ended up sharing data and technologies.
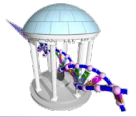And the competition led to a completed draft 5 years ahead of schedule.

The Sequencing Race
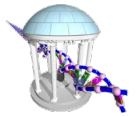
Since the Human Genome project there have been an explosion of genomes sequenced. Initially, the focus was on model organisms, then favorites, then all of human diversity, and finally a catalog of life's diversity.
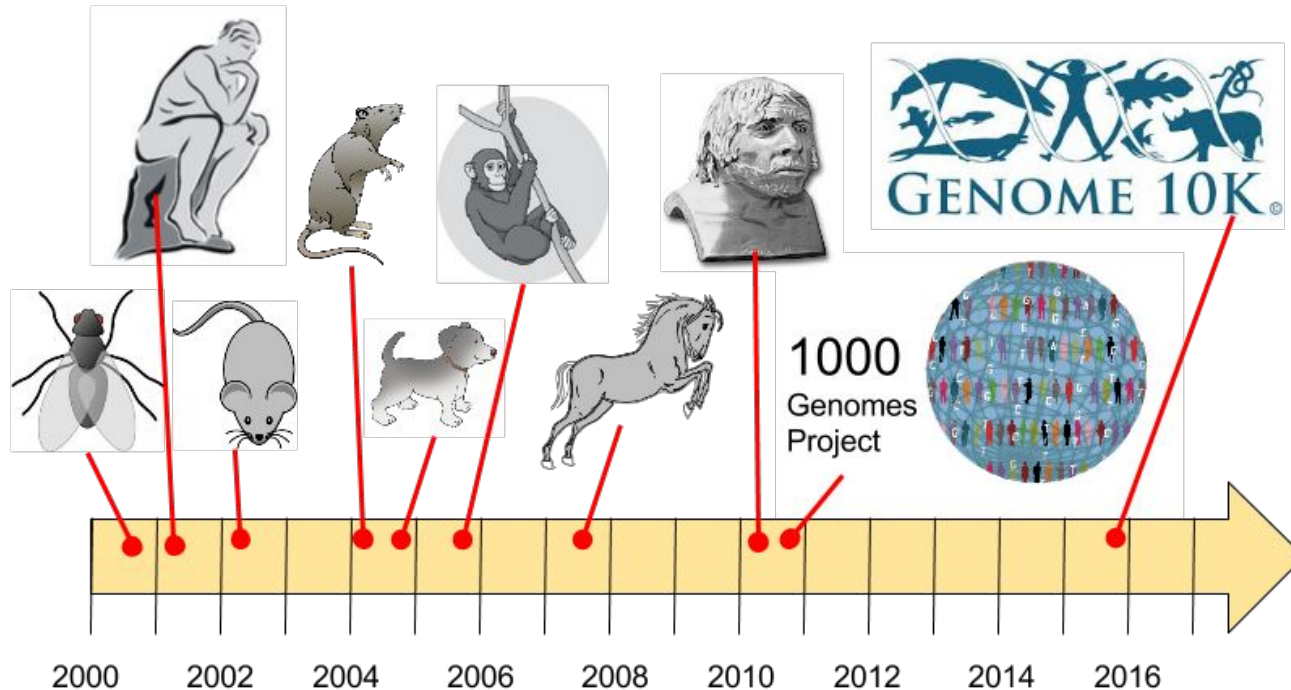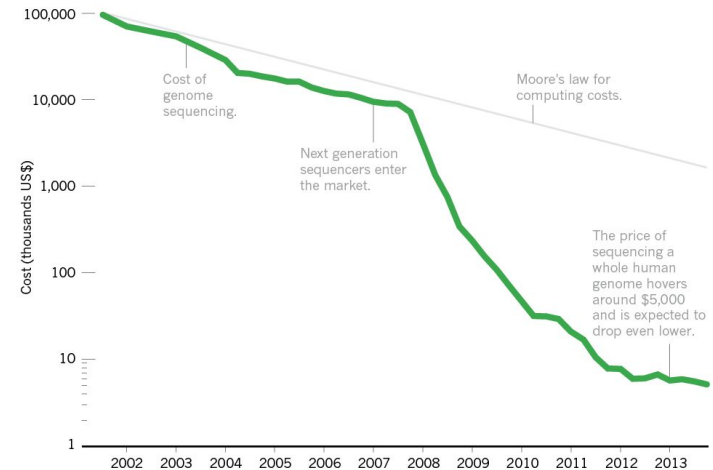
The secret behind this explosion of genomes

Next generation sequencing machines have revolutionized the DNA sequencing process. They work in various ways including massiviely-parallel single-base extension methods, to captured Dnases whose motions suggest a the base being replicated, to microholes that only a single DNA molecule can pass through, and the bases are determined by detectable charge differences.

In a way, the *genome moonshot* was far more successful than the real moonshot. The rate at which genomes can be sequenced, and the cost per base has seen unprecented improvements. Faster than even Moore's Law.

Falling fast

In the first few years after the end of the Human Genome Project, the cost of genome sequencing roughly followed Moore's law, which predicts exponential declines in computing costs. After 2007, sequencing costs dropped precipitously.
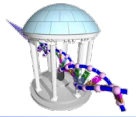
Cost of genome sequencing.

Moore's law for computing costs.

Next generation sequencers enter the market.

The price of sequencing a whole human genome hovers around $5,000 and is expected to drop even lower.

How does it all work?

Multiple Copies of a Genome

Reads

High Coverage **Low Coverage**

Consensus Sequence
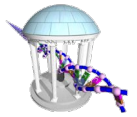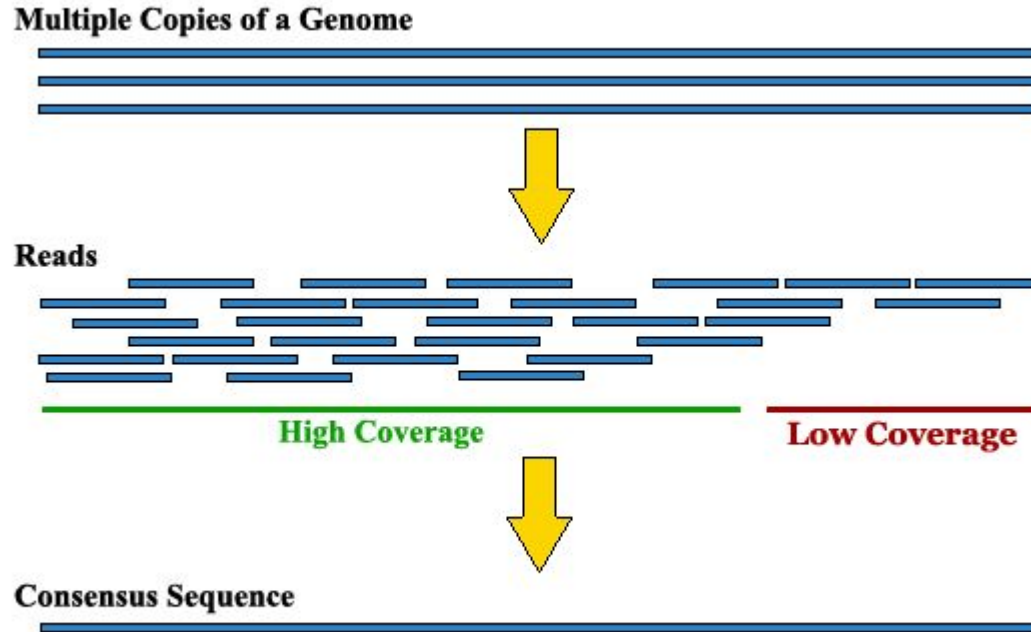
It is as if we must first smash a grecian urn in order to completely see it.

An Analogy
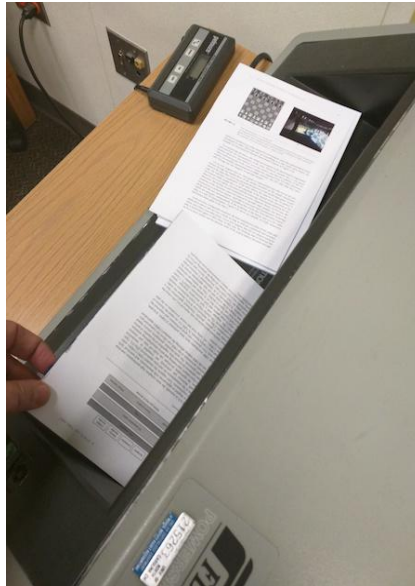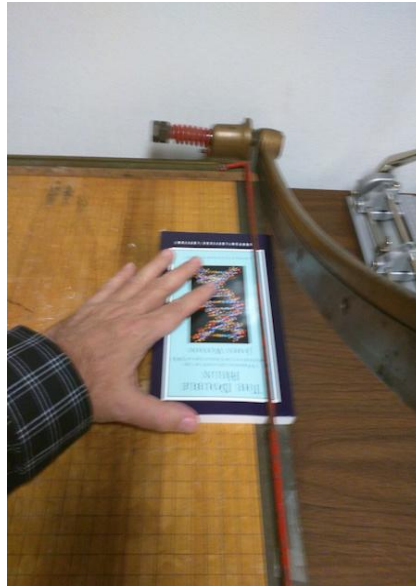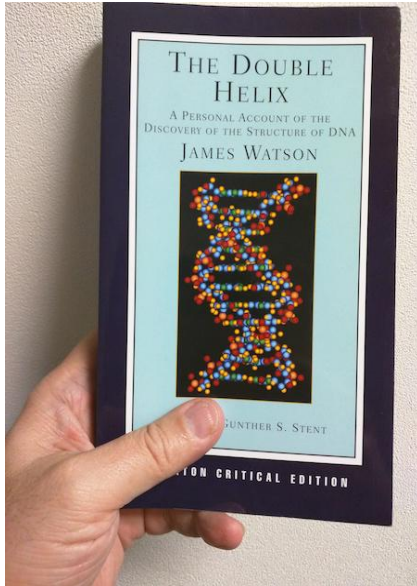
Some important differences

- A better analogy would have been to shred 100's of books
- Shuffle the pages before shredding
- Oh yeah, my book has approximately 850,000 characters.
- The entirety of Encyclopedia Britannica is approximately 250,000,000 characters.
- Your genome is approximately 12 times larger
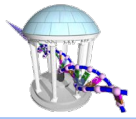
How would you Reassemble our Book?

Each paper shed is like a DNA fragment, or read.

Searching for overlaps
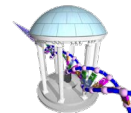
You'd look for fragments that fit together based on some overlapping context that they share.

our way to Soho for s
too long over his mistake might be fatal.
wrong instead of looking like a fool. So
was the further danger that if he put one

And then, build upon those to assemble a more complete picture.

ll out of the reach of Francis and me.
our way to Soho for s   turned to the problem of Linus, emphasizing that smiling
too long over his mistake might be fatal. The position would be far safer if Pauling had been merely
wrong instead of looking like a fool. Soon, if not already, he would be at it day and night. There
was the further danger that if he put one of his assistants to taking DNA photographs, the B.

Finally you assemble a *nearly* complete version
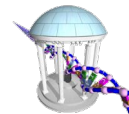
- How can we code such an approach?

- What is *overlapping context* in our DNA fragments?

- How would we represent and manage these overlaps?

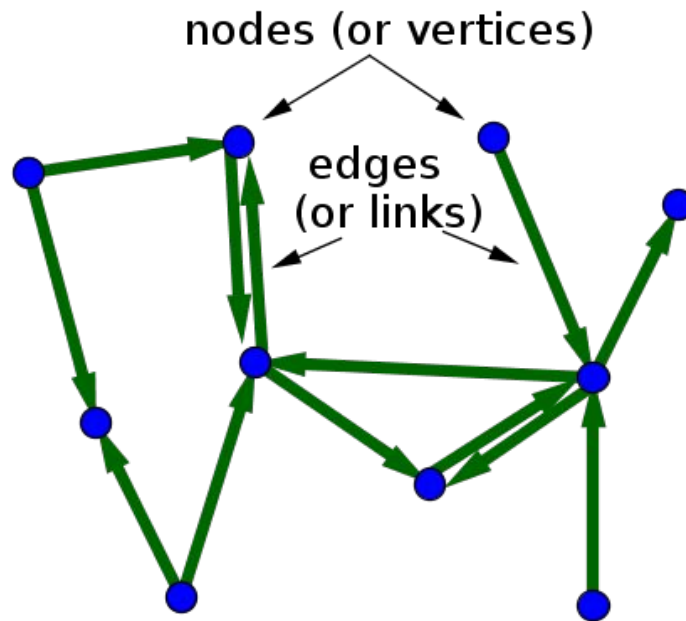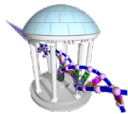# Key idea: Finding links between pairs
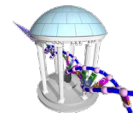
This leads us to a computational analogy called a graph

A graph is composed of nodes, which can represent entities, in our case read fragments

Nodes are connected by edges that represent some relationship between a pair of nodes

The edges of a graph can be directed

nodes (or vertices)

edges (or links)

De Bruijn's Problem and his Graphs

Nicolaas de Bruijn (1918-2012)

A dutch mathematician noted for his many contributions in the fields of graph theory, number theory, combinatorics and logic.

Minimal Superstring Problem:

Find the shortest sequence that contains all $|\Sigma|^k$ strings of length k from the alphabet Σ as a substring.

Example: All strings of length 3 from the alphabet {'0','1'}.

binary3 = {'000', '001', '010', '011', '100', '101', '110', '111'}

```
                        101 100                  111 100
                        001 111                  001 101
Solution #1: 0001011100        Solution #2: 0001110100
                        000 011                  000 110
                        010 110                  011 010
```

He solved this problem by mapping it to a graph. Note, this particular problem leads to cyclic sequence.

Construct a "graph" of a sequence
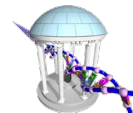
For the moment let's imagine that reads are like k-mers from a sequence, as they do tend to be uniform in length.

```
GACGGCGGCGCACGGCGCAA        - Our toy sequence
GACGG
 ACGGC
  CGGCG
   GGCGG
    GCGGC
     CGGCG
      GGCGC              - The complete set of 16 5-mers
       GCGCA
        CGCAC
         GCACG
          CACGG
           ACGGC
            CGGCG
             GGCGC
              GCGCA
               CGCAA
```
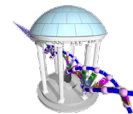
Now we can construct a graph where:
1. Each 5-mer is a node
2. There is a directed edge from a k-mer that shares its (k-1)-base suffix with the (k-1)-base prefix of another

A read-overlap graph

The read-overlap graph for the 5-mers from:

GACGGCGGCGCACGGCGCAA

The problem is:

**How to infer the original sequence
From this graph?**

Our original sequence is
just a path in this graph.
How would you find it?

Parlor games

Once finding paths in graphs was a popular form of entertainment…
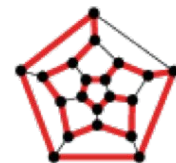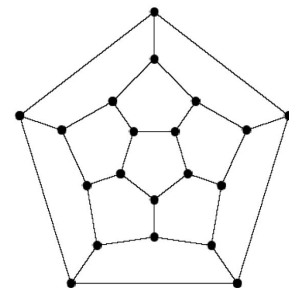
Graphs would be printed in newspapers, and people would try to find paths in them as a game.
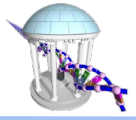
The rules of our game

- Every node, k-mer, can be used exactly once
- The object is to find a path along edges that visits every node one time
- This game was invented in the mid 1800's by a mathematician called **Sir William Hamilton**

An example of Hamilton's game:
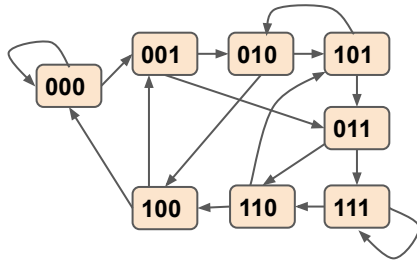
Finding a Hamiltonian Path in our graph

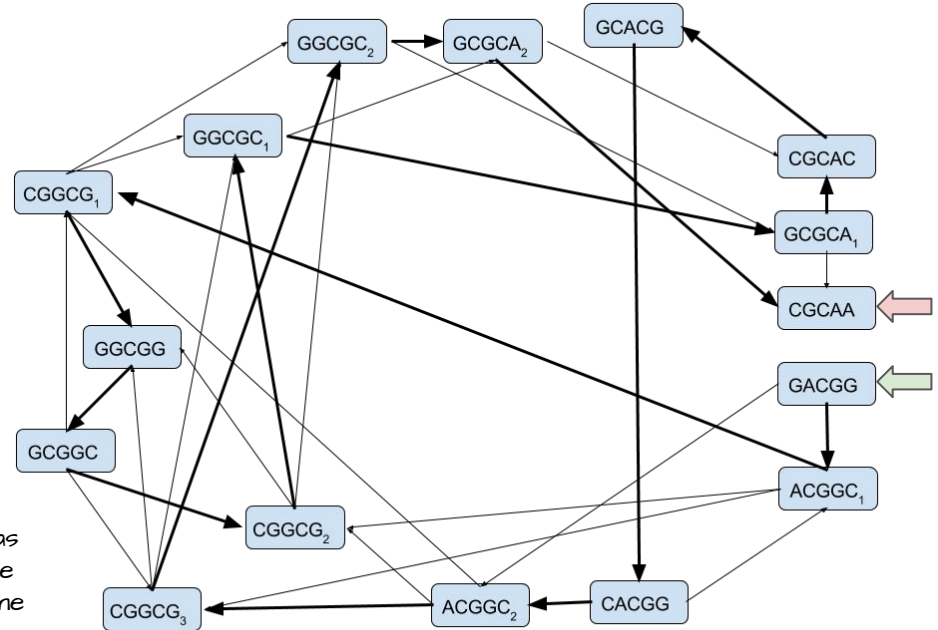For our desired sequence:

GACGGCGGCGCACGGCGCAA

is indeed a path in this graph.

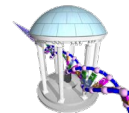How would you write a program
To solve Hamilton's puzzles?

Is the solution unique?

de Bruijn knew this was
a hard problem, But, he
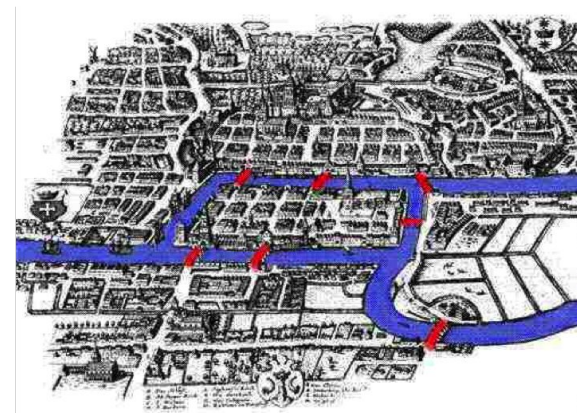also knew another game
he could play.

Euler's Tour

The rules of a new game

- Every *edge*, k-mer, can be used exactly once
- The object is to find a path in the graph that uses each *edge* only one time
- This game was invented in the late 1700's by a mathematician called Leonhard Euler
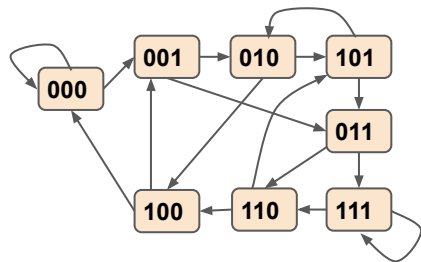
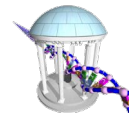Leonhard Euler

A version of Euler's game:

Bridges of Königsberg: Find a city tour that crosses every bridge just once

001 → 010 → 101

000

011

100 ← 110 ← 111

How can I make my "k-mers" into edges rather than nodes?
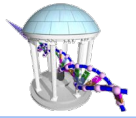
Another representation of k-mers in a graph

- Rather than making each k-mer a node, let's try making them an edge
- That seems odd, but it is related to the overlap idea
  - The 5-mer GACGG has a prefix GACG and a suffix ACGG
  - Think of the k-mer as the edge connecting a prefix to a suffix
  - This leads to a series of simple graphs
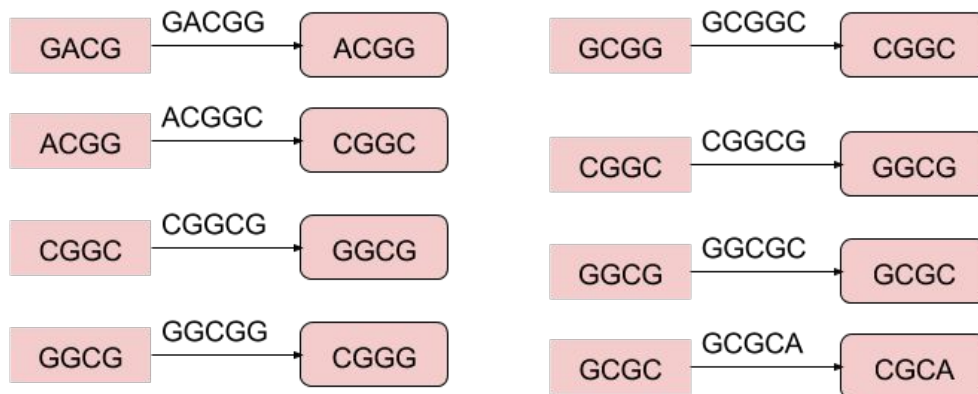
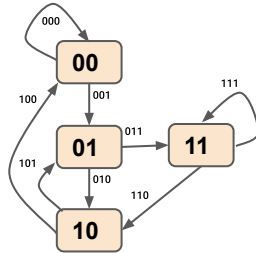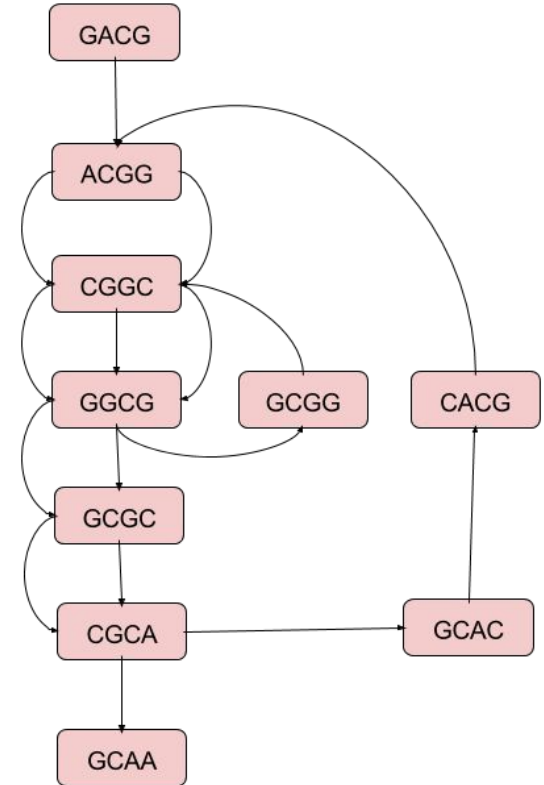 - Then combine all nodes with the same label

A De Bruijn Graph

This graph, like the previous one has the property that edges connect nodes where a k-1 suffix matches a k-1 prefix. Graphs of this type are called "De Bruijn" graphs, after a famous mathematician.

Recall that our original 5-mers are edges in this graph, whereas they were nodes in the previous one.

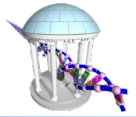Now, how might you infer the original sequence using this graph?

Some of those bridges leave and come back to the same island.

Two graphs, same problem

Two graphs representing 5-mers from the sequence "GACGGCGGCGCACGGCGCAA"

Hamiltonian Path:

Each k-mer is a vertex. Find a path that passes through every *vertex* of this graph exactly once.

Eulerian Path:

Each k-mer is an edge. Find a path that passes through every *edge* of this graph exactly once.

Next Time

- Code to solve our graph problems

- Code that is simple

- Code that is fast