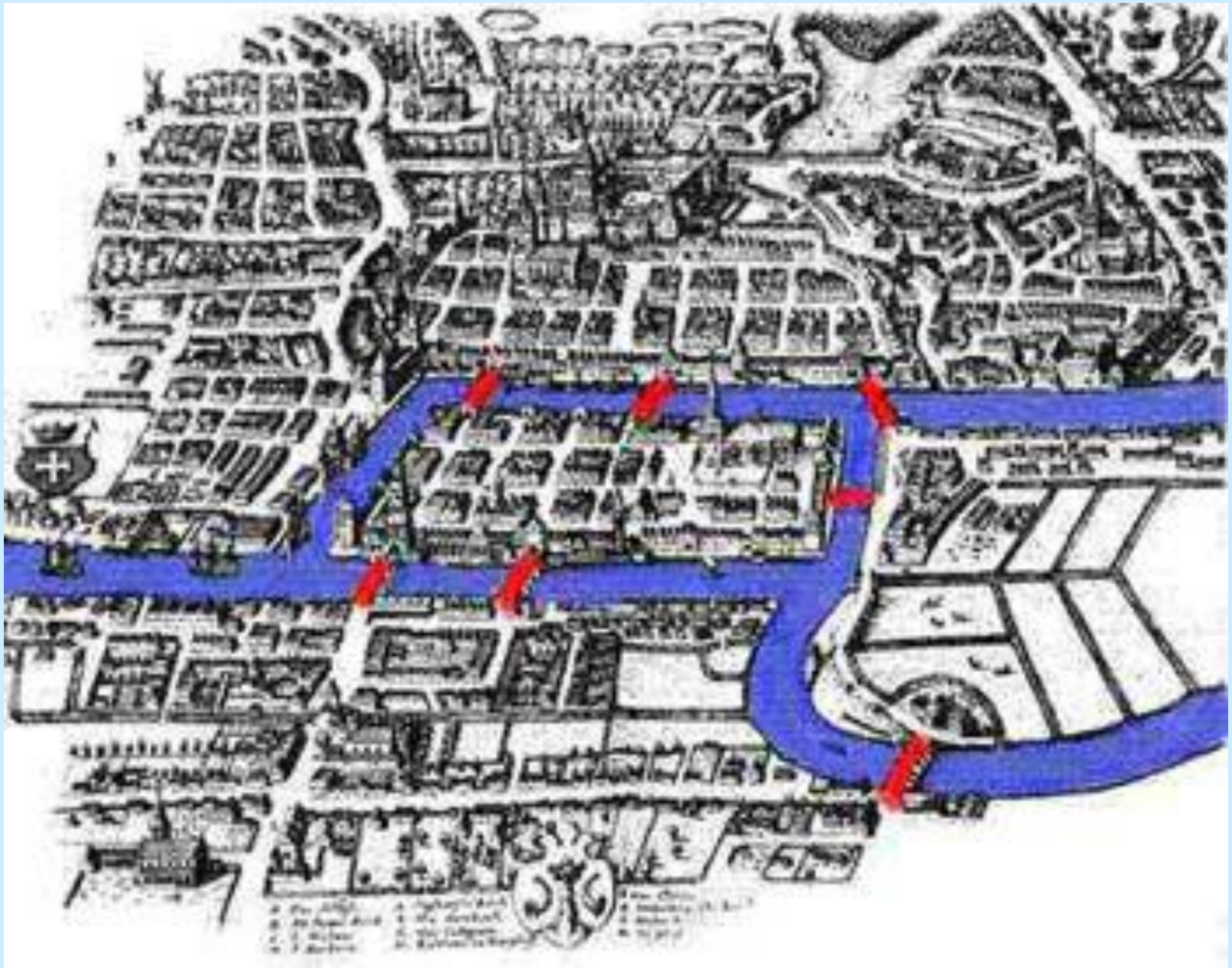
IE 102 Fall 2019

Routing Through Networks - 1

The Bridges of Koenigsberg: Euler 1735

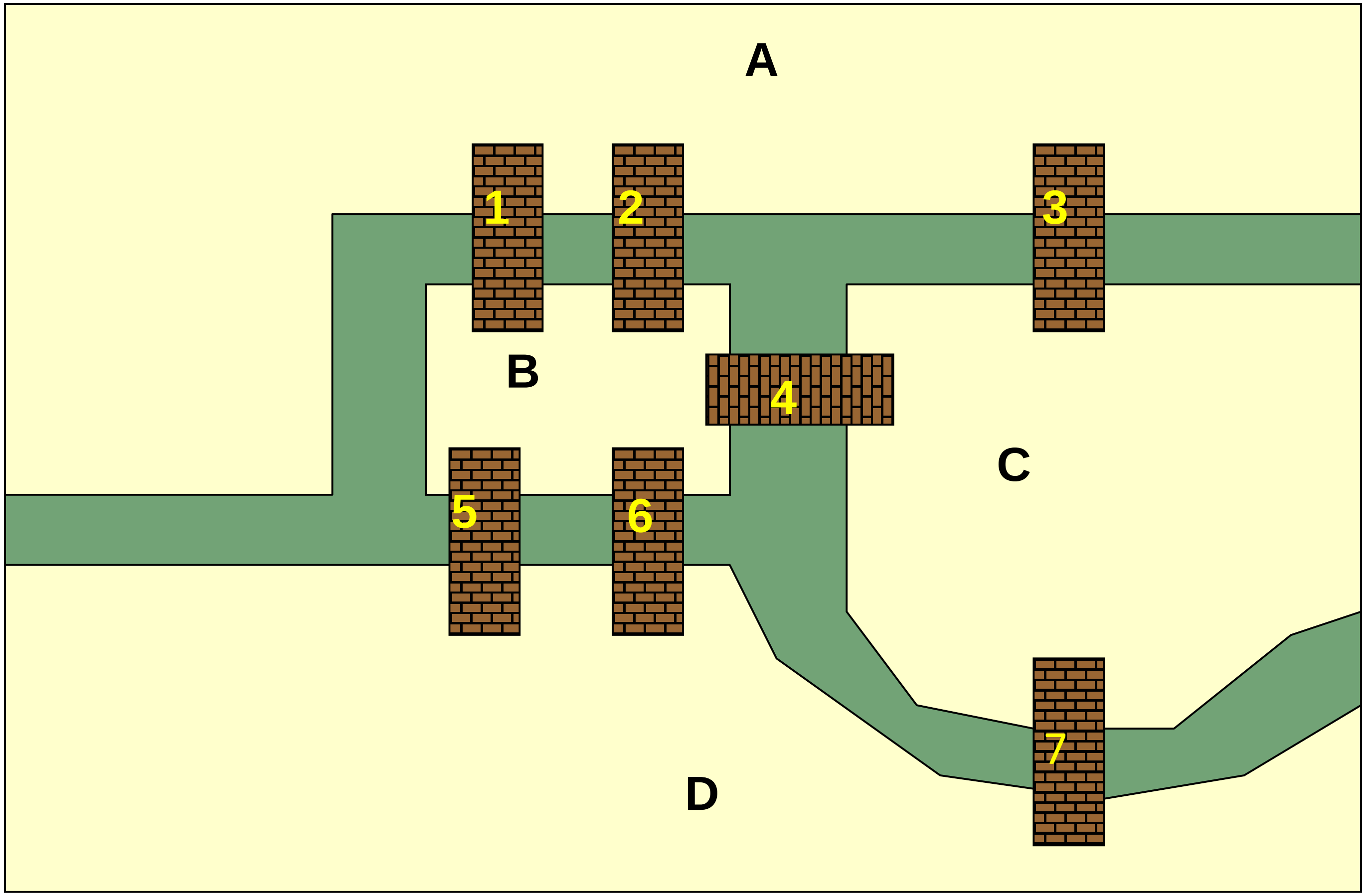
- “Graph Theory” began in 1735
- Leonard Eüler
  - Visited Koenigsberg
  - People wondered whether it is possible to take a walk, end up where you started from, and cross each bridge in Koenigsberg exactly once
  - Generally it was believed to be impossible


Circle
NORWAY
SWEDEN
FINLAND
Oulu
Umeå
Gulf of Bothnia
Trondheim
Tampere
Helsinki
Gävle
Gulf of Finland
St. Petersburg
Lake Onega
Lake Ladoga
Bergen
Oslo
Stockholm
Åland Islands
Tallinn
ESTONIA
RUSSIA
Orkney Islands
Skagerrak
Göteborg
Gotland
LATVIA
Riga
North Sea
Baltic Sea
Öland
DENMARK
Copenhagen
LITHUANIA
Smolensk
Newcastle
Bornholm
Kaliningrad
RUSSIA
Vilnius
Minsk
BELARUS
Gdańsk
Rostock
Hamburg
Berlin
Amsterdam
Poznań
Warsaw
Brest
Kiev
NETHERLANDS
Hannover
POLAND
GERMANY
Leipzig
Wrocław
Brussels
Bonn
Kraków
BELGIUM
Frankfurt
Prague
L'viv
UKRAINE
Le Havre
CZECH REPUBLIC
Luxembourg
Paris
LUX.
SLOVAKIA
MOLDOVA
Chişinău
Stuttgart
Bratislava
Strasbourg
Odesa
Budapest
Munich
Vienna
Cluj-Napoca
AUSTRIA
FRANCE
SWITZ.
LIECH.
Graz
HUNGARY
ROMANIA
Bern
Geneva
Pécs



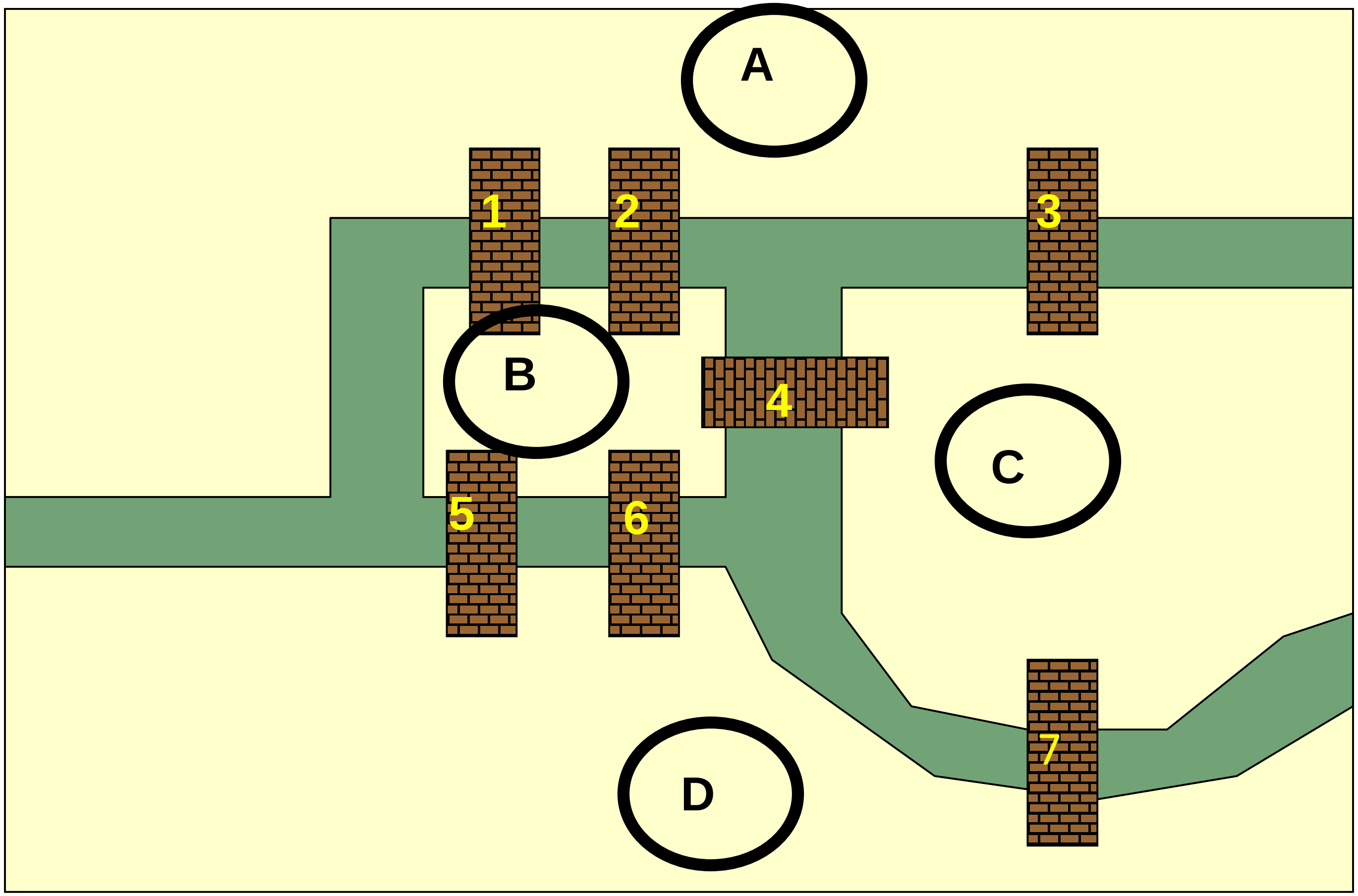

Königsberg i. Pr.
Schloßteichbrücke

The Bridges of Koenigsberg: Euler 1736



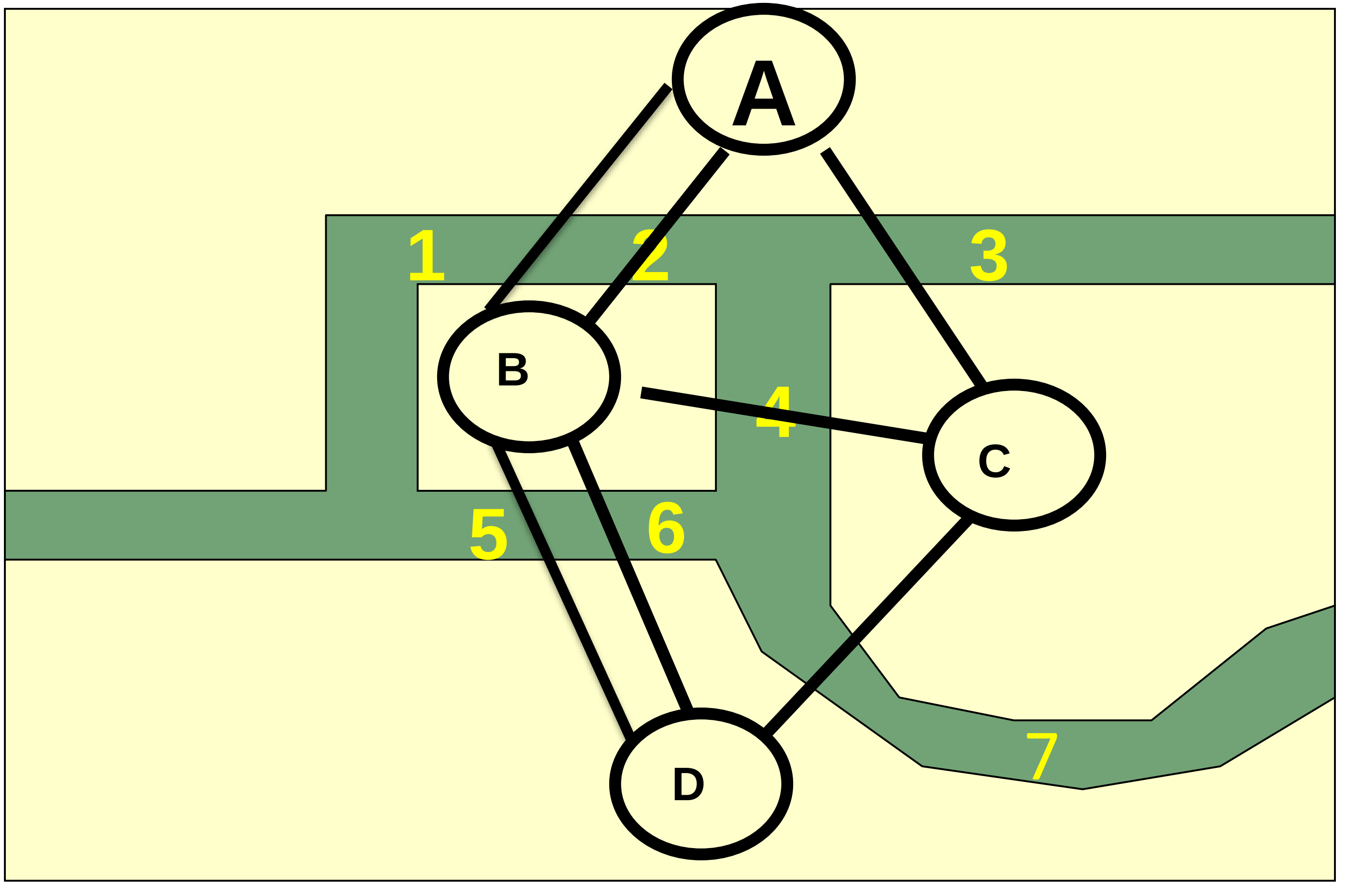
Is it possible to start in A, cross over each bridge exactly once, and end up back in A?

The Bridges of Koenigsberg: Euler 1736



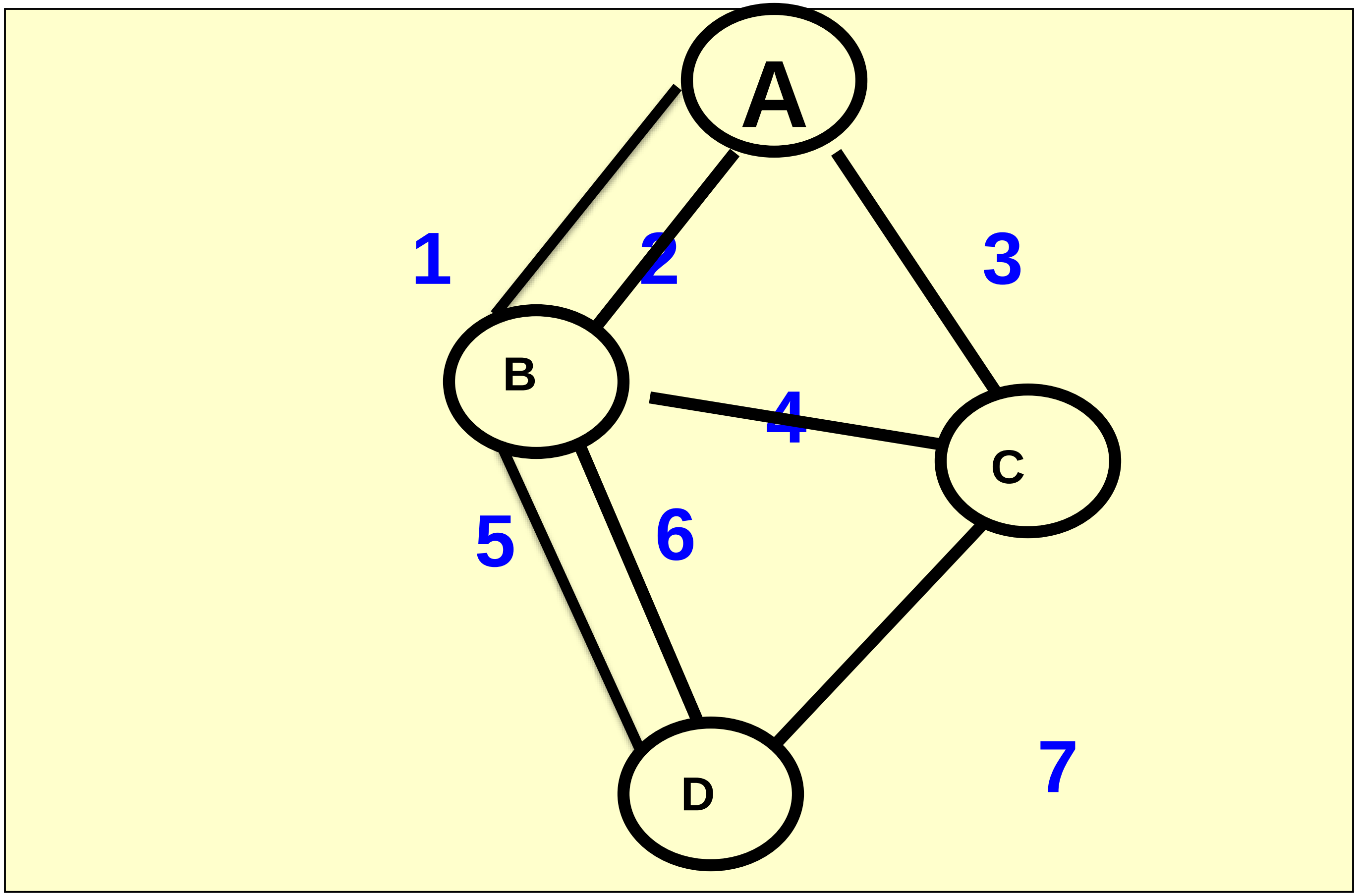
Conceptualization: Land masses are “nodes”.

The Bridges of Koenigsberg: Euler 1736



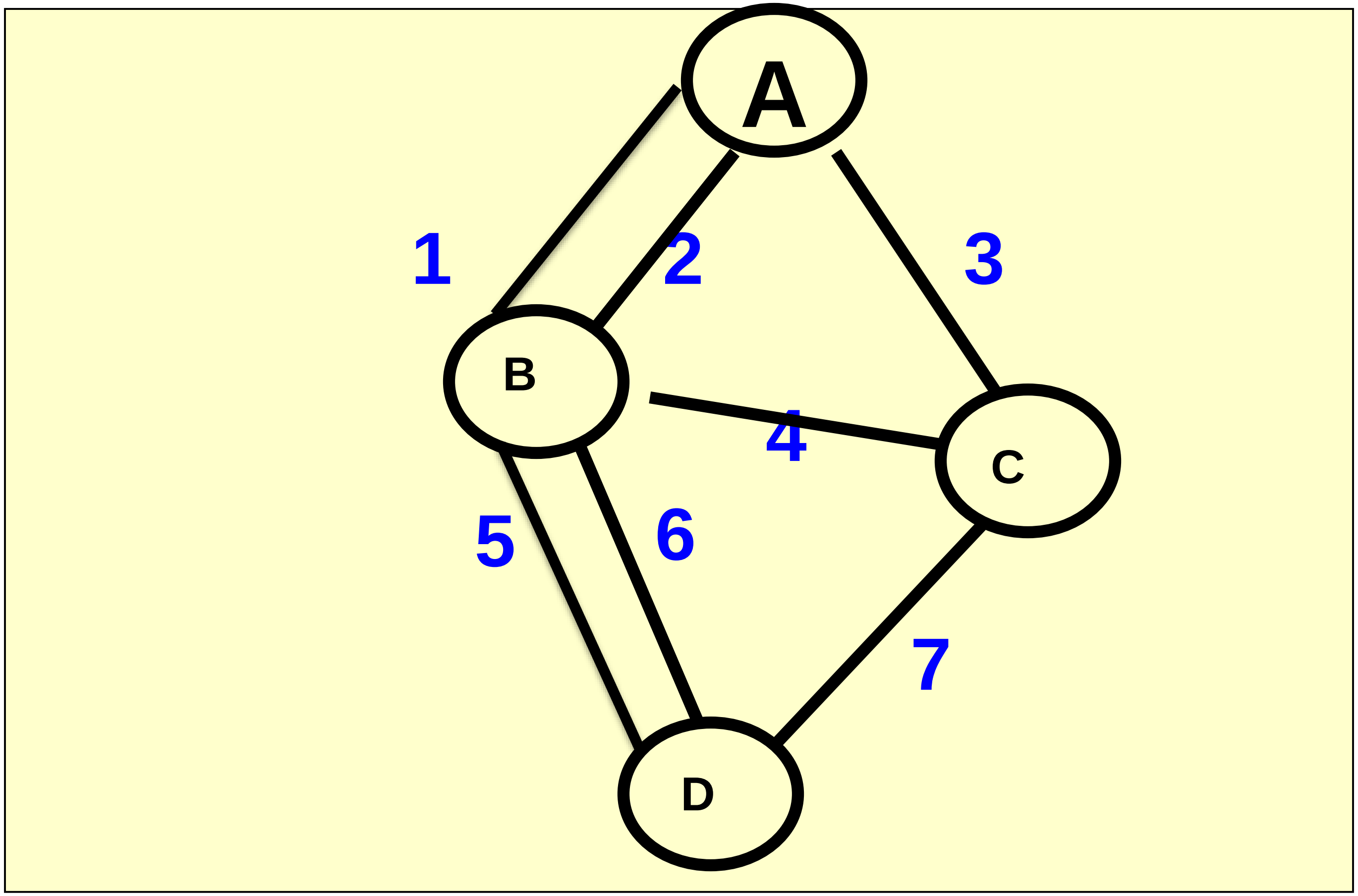
Conceptualization: Bridges are "arcs."

The Bridges of Koenigsberg: Euler 1736



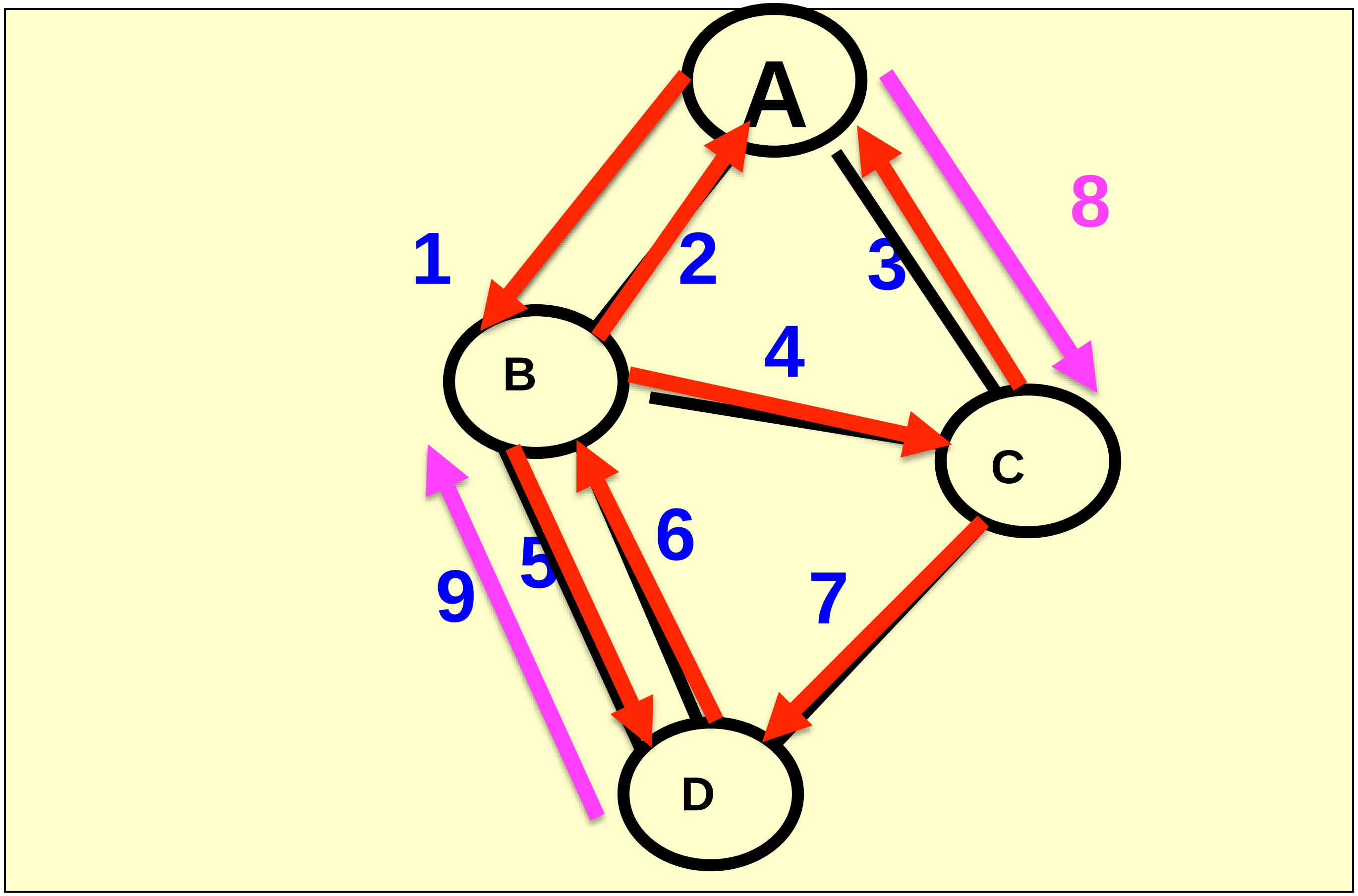
Is there a "walk" starting at A and ending at A and passing through each arc exactly once?

The Bridges of Koenigsberg: Euler 1736



Is there a closed tour starting at A and ending at A and passing through each arc exactly once?
Such a walk is called an *eulerian cycle*.

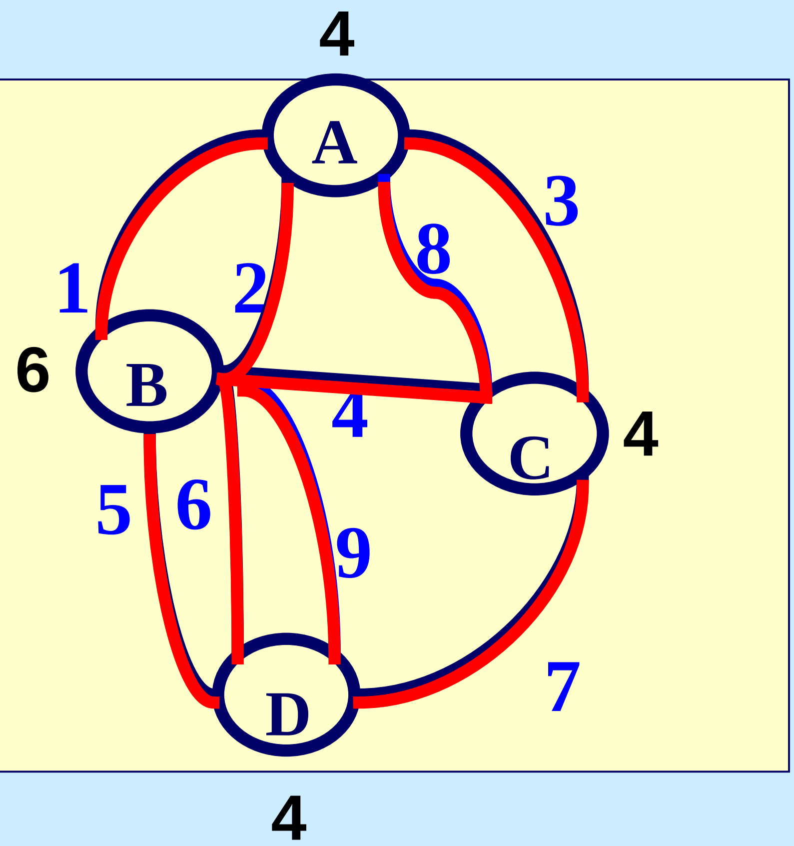
Adding two more bridges creates such a walk



Here is the tour

A, 1, B, 5, D, 6, B, 4, C, 3, A, 8, C, 7, D, 9, B, 2, A

On Eulerian Cycles



The degree of a node in an undirected graph is the number of incident arcs

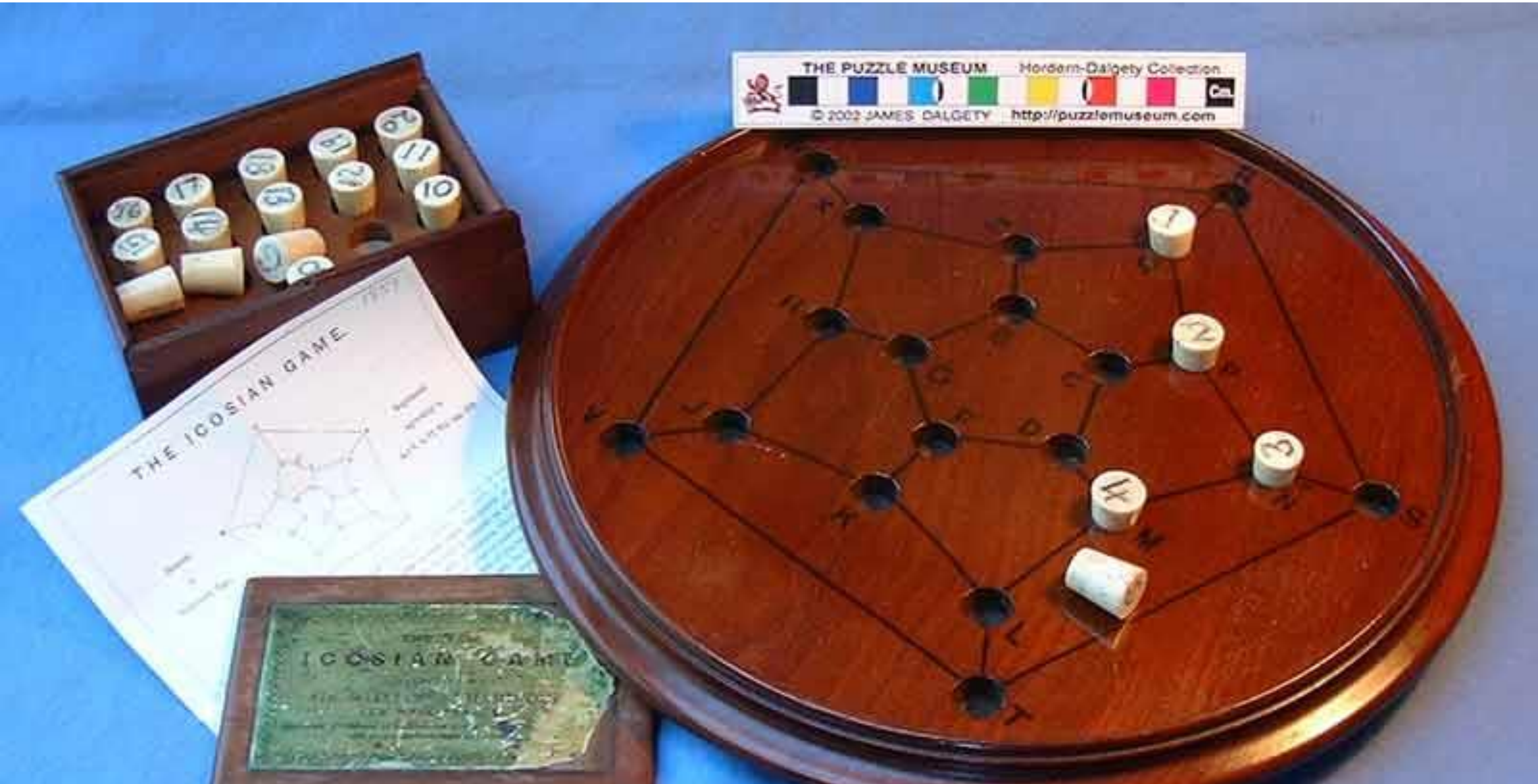
Theorem. An undirected graph has an eulerian cycle if and only if

(1) every node degree is even and

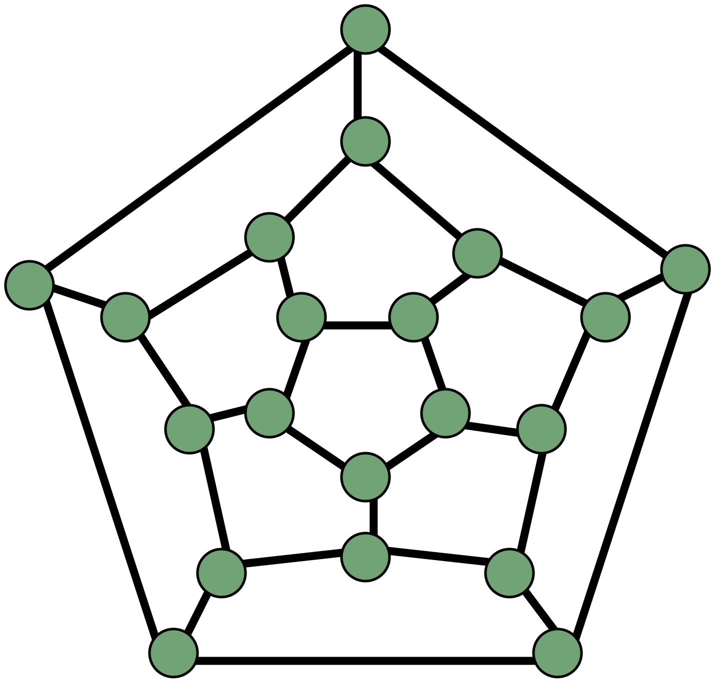
(2) the graph is connected (that is, there is a path from each node to each other node).

Hamilton's Around the World Game

In 1857, the Irish mathematician, Sir William Rowan Hamilton invented a puzzle that he hoped would be very popular.



Hamilton’s Around the World Game

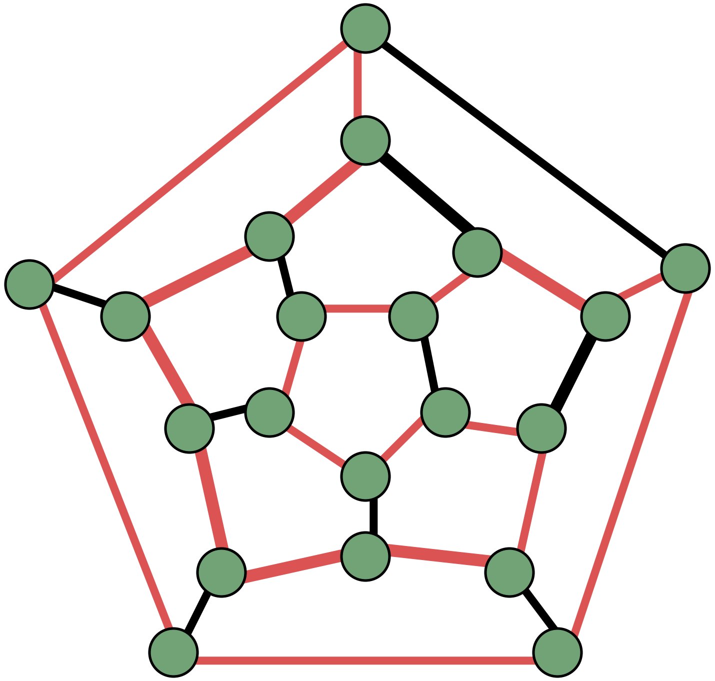


The objective was to make what we just called a hamiltonian cycle.

The game was not a commercial success, especially the 3D version.

But the mathematics of hamiltonian cycles is very popular today.

Hamilton's Around the World Game



We will see this problem again when we generalize it to be the *traveling salesman problem.*

Where Network Flows Arise

- Transportation
  - Transportation of goods over transportation networks
  - Scheduling of fleets of airplanes: time/space networks
- Manufacturing
  - Scheduling of goods for manufacturing
  - Flow of manufactured items within inventory systems
- Communications
  - Design and expansion of communication systems
  - Flow of information across networks
- Personnel Assignment
  - Assignment of crews to airline schedules
  - Assignment of drivers to vehicles

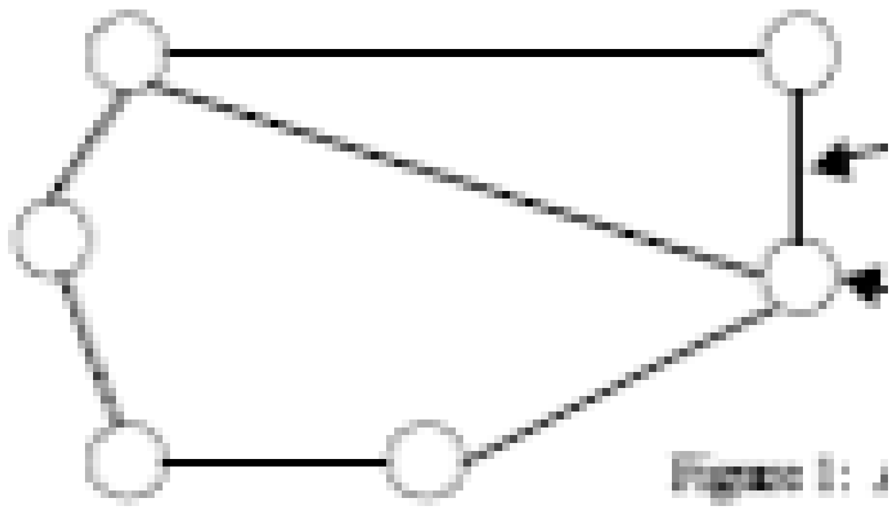
Travel and Delivery Problem

- Aim: To go from one-location to another

- Repetitive operations
- Example 1 – Kebab Delivery
- Example 2 – Mail Delivery
- Example 3 – Goods Delivery for a Retailer
- Example 4 – Electricity-Meter Reading

Representation by networks

- Graphical representation
- Nodes (vertices) + arcs (edges)



Representation by networks

- Nodes – often represent geographic points (cities, intersections, railroad stops, production points, ...)
- Arcs – often represents links between nodes (Roads, railroads, ...)
- Arcs can be undirected (two-way) or directed (one-way)
- Nodes, arcs can be uncapacitated, capacitated
- Arcs are weighted with numerical values (distance, time, cost, ...)

Hamiltonian Path Problem

- Assume a network with all possible arcs – called complete graph
- Hamiltonian Circuit (path, graph) – a path that visits each node exactly once
- Find the path that minimizes total “distance” written on the arcs.
- An approach: Brute-force approach (total enumeration) Try every unique circuit

Questions to answer:

- How many possible circuits?
- Take any circuit and travel in reverse
- How many unique circuits remain?
- Writing General formulas:
- How many total circuits? Without duplicates?

More questions to answer:

- Let n = 21. Then total number of unique circuits without repetition 20!/2 – 1.22×10^{18}
- What is an operation?
- Let the fastest computer do approximately 1 trillion (1×10^{12}) operations (computations) per second
- Let us assume that we need more or less 100 operations to obtain a circuit
- 1400 days if the computer runs continuously

Traveling Salesman Problem - TSP

- Definition: What is the shortest possible route that visits each city exactly once and returns to the origin city?
- Find the Hamiltonian path that minimizes total “distance” written on the arcs.

More on solution

What if enumeration is not possible?

- Will they give the optimal solution? Not always
- A good solution? Not sure
- But can be applied in reasonable time
- We can try a bit more sophisticated techniques, as well.
- These methods are called “heuristics”.

Ending Remarks

- NP-Hard (non-polynomial) - # of operations required by the solution algorithm is not polynomial in # of nodes.
- NP-Hard problems can not be solved to optimality unless # of nodes is “reasonable”.
- We resort to use of heuristics – no quarantee of optimality; if lucky get bounds – benchmark.

Ending Remarks (continued)

❑ Heuristics

- Should be easier to apply
- Should be based on some “logic”

➢ Construction heuristic – construct a solution (Random, Greedy / Nearest Neighbour)

➢ Improvement heuristic – improve an available solution (2-opt)

➢ You need both aspects, in general.

❑ Implications for IE curriculum

Heuristics

- Random
  - Connect the cities randomly
- Nearest Neighbour
  - Each time insert the closest unvisited node
- Greedy
  - Go along the cheapest arc between a visited and unvisited node, continue until none remains

- Then apply 2-opt
  - Choose two arcs and switch if the distance decreases. Continue until no such switch is possible

Heuristics

Videos:

1) Solving the TSP with Heuristics

2)TSP-Random, Greedy, 2-Opt

3) TSP-Visualization

KAHOOT!

Go to kahoot.it

Enter nickname as your ID.