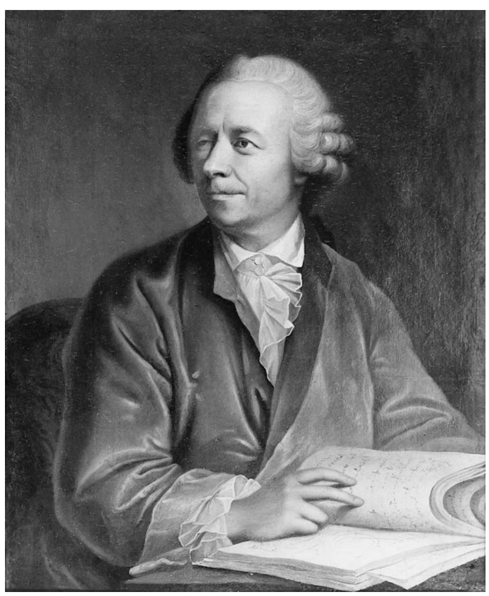
Week 3: Eulerian and Hamiltonian Cycles

August Deer, Siddarth Chalasani, Oleg Gleizer

July 17, 2022

Eulerian paths and cycles

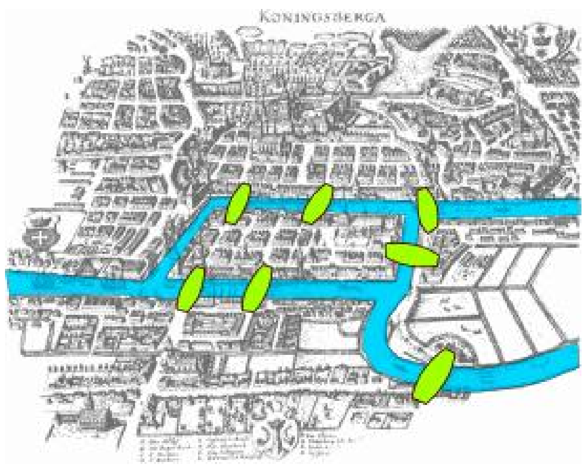
An *Eulerian path* is a path in an (undirected) graph that traverses each edge exactly once. An *Eulerian cycle* is a closed Eulerian path. They are named in honour of a great Swiss mathematician, Leonhard Euler (1707-1783), considered by many as the founding father of the graph theory.



Leonhard Euler

The seven bridges of Königsberg problem

During his stay in the city of Königsberg, then the capital of Prussia, Euler came up with, and solved, the following problem. Can one design a walk that crosses each of the Königsberg's seven bridges once and only once? The picture of Königsberg of Euler's time is provided below.



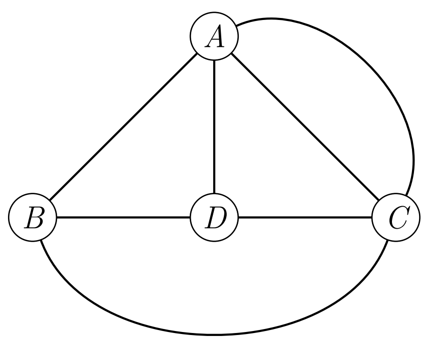
Map of Königsberg in Euler's time showing the actual layout of the seven bridges, highlighting the river Pregel and the bridges.

Problem 1. *Draw a graph with the vertices corresponding to the landmasses from the picture above and with the edges corresponding to the Königsberg's seven bridges. What are the degrees of each of the graph's vertices?*

Problem 2. *Was it possible to design a Eulerian walk in the city of Königsberg at the time of Euler? Why or why not?*

Problem 3. *Can you add a bridge so a walk becomes possible?*

Problem 4. *Find a Eulerian path in the following graph.*



Problem 5. *Does the above graph contain a Eulerian cycle? If not, how many edges do you need to add to create an Eulerian cycle?*

Problem 6. *What is the minimal number of times you need to lift the pencil off the paper to draw a cube without repeating any edge?*

Challenge Problems

Problem 7. *Can you add a bridge to the map of Königsberg without creating an Eulerian walk?*

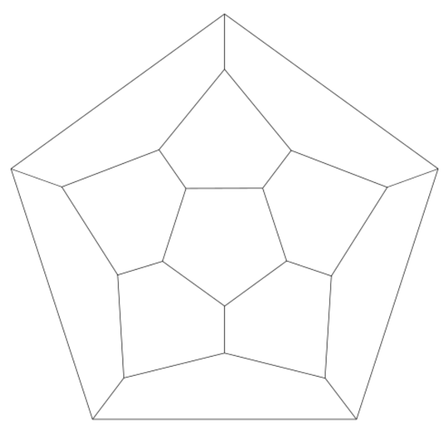
Problem 8. *Given any graph, how can you tell whether an Eulerian cycle is possible? (Hint: Think about the degrees of all the vertices.)*

Hamiltonian paths and cycles

A *Hamiltonian path* is a path in a graph that visits each vertex exactly once. A *Hamiltonian cycle* is a closed Hamiltonian path. They are named in honour of the great Irish mathematician, physicist, and astronomer, Sir William Rowan Hamilton (1805-1865).

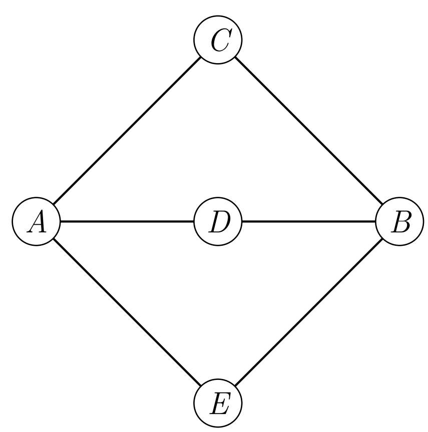
Sir William Rowan Hamilton

Problem 9. *Sir Hamilton sold a puzzle in which you had to traverse a dodecahedron along the edges, visit every vertex exactly once, and return to where you started. Solve the puzzle. In other words, find a Hamiltonian cycle on the following graph:*



Problem 10. *A graph with V vertices contains a Hamiltonian cycle. What is the minimal number of its edges?*

Problem 11. *Prove the following graph has no Hamiltonian cycle.*

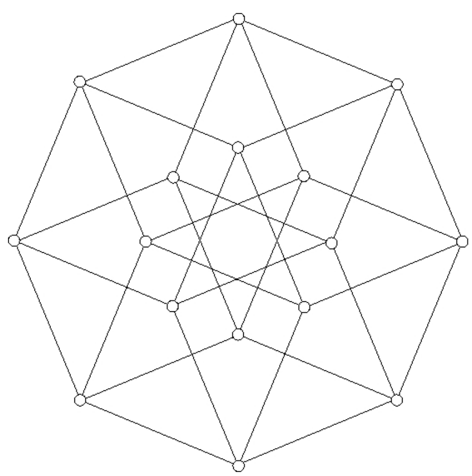


Theorem 1. *(Ore) Let G be a connected graph with $n \geq 3$ vertices. If $deg(u) + deg(v) \geq n$ for every pair of non-adjacent vertices u and v, then G is Hamiltonian.*

Problem 12.

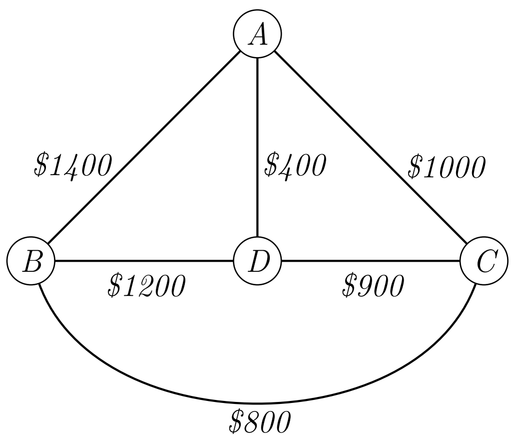
- *Does the graph below satisfy the conditions of Ore's theorem?*

- *Find a Hamiltonian cycle in the graph.*



Travelling salesman problem

Problem 13. *A salesman with the home office in Albuquerque has to fly to Boston, Chicago, and Denver, visiting each city once, and then to come back to the home office. The airfare prices, shown on the graph below, do not depend on the direction of the travel. Find the cheapest way.*



Problem 13 is a simple case of the *travelling salesman problem* (TSP). Let G be a graph, directed or undirected, with vertices $v_1, v_2, \ldots, v_n$. Its edges (v_i, v_j) are *weighted* – have numbers assigned to them. The TSP is to find a Hamiltonian cycle of lowest weight. The TSP appears in areas as different as scheduling, microchip design, DNA sequencing, and more.

Challenge Problems

Problem 14. *Consider the continental United States (i.e., without Hawaii and Alaska) and view it as a graph in the following manner:*

1. *The vertices are the states*

2. *Two vertices share an edge if the states share a border*

Notice that Maine only borders one state, so any Hamiltonian path would have to either start or end in Maine. Suppose we start in Maine. New York is what's called a **bottleneck***; removing New York would disconnect the graph. Therefore, any Hamiltonian path which starts at Maine would have to end on the other side of New York (which we will call the Western US). An all knowing computer has verified that there are Hamiltonian paths ending at every single state west of New York* **except one***. Which state is the exception?*

Shortest Path Algorithms

A problem related to the Traveling Salesman is the *Shortest Path Problem*, which asks for the path with least weight from vertex a to vertex b. Unlike the Traveling Salesman problem, which has no known fast solution, there are many algorithms for quickly finding the shortest path in a weighted graph. One such algorithm was created by Edsger W. Dijkstra. Dijkstra's algorithm goes as follows:

Dijkstra's Algorithm. Let $L(x)$ be the distance from vertex a to vertex x. The following steps will give us $L(b)$

1. Let $L(a) = 0$, and $L(x) = \infty$ for all other x. These values will be refined as the process goes on. Initialize set $T = \mathcal{V}$, the set of all vertices.

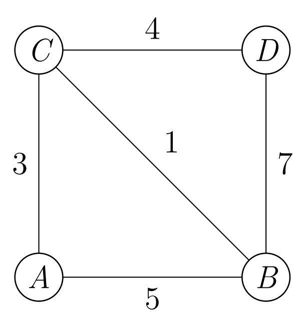
2. While vertex b is in T:

   (a) Set v to be a vertex in T with the minimum $L(v)$.

   (b) Remove v from set T.

   (c) For each vertex x in T adjacent to v, set $L(x)$ to the minimum of $L(x)$ and $L(v) + w(v, x)$, where $w(v, x)$ is the weight of the edge connecting v and x.

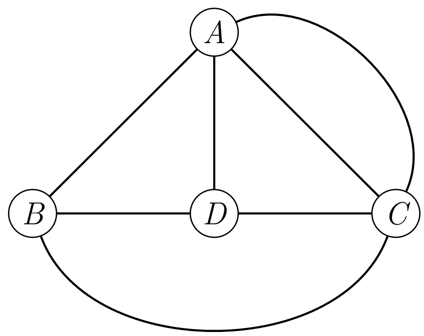
Problem 15. *Use Dijkstra's Algorithm to find the shortest path from vertex A to B in the following graph.*



Minimum Spanning Tree

Another related problem is the Minimum Spanning Tree problem. A *tree* is a graph without any cycles. This means that there is a unique path between any two vertices. Given a graph G with vertices $\mathcal{V}$ and edges $\mathcal{E}$, a *spanning tree* of G is a connected subgraph of G with the same vertex set $\mathcal{V}$, and without any cycles.

Problem 16. *Find a spanning tree of the following graph.*



Given a weighted graph G, the *minimum spanning tree* (MST) of G is a spanning tree of G with minimum total edge weight. There are multiple algorithms for efficiently finding the minimum spanning tree of a weighted graph. One such algorithm is Prim's algorithm, which goes as follows:

Prim's Algorithm. Let $C(x)$ be the cost of adding vertex x to the MST and $E(x)$ be the edge used to add vertex x to the MST. The following steps will give us the MST:

1. Label any vertex v_0 as the first vertex added to the MST. Let $C(v_0) = 0$, and $C(x) = \infty$ for all other vertices x. Let $E(x)$ be empty for all vertices x. Initialize set $Q = \mathcal{V}$, the set of all vertices.

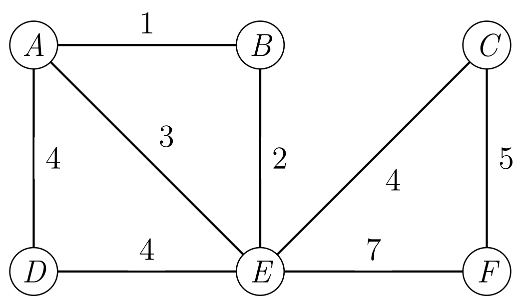
2. While Q is non-empty:

    (a) Remove vertex v from Q with minimum value of $C(v)$.

    (b) Add vertex v and edge $E(v)$ (if $E(v)$ is not empty) to the MST.

    (c) For each vertex x in Q that is adjacent to v, if $w(v, x) < C(x)$, set $C(x) = w(v, x)$ and $E(x) = \{v, x\}$.

Problem 17. *Use Prim's Algorithm to find a minimum spanning tree in the following graph.*



Problem 18. *Is the path between two vertices on an MST always the same as the shortest path on the original graph?*

Challenge Problems

Problem 19. *Prove that Dijkstra's algorithm gives the shortest path between two vertices on a weighted graph.*

Problem 20. *Prove that Prim's algorithm gives the minimum spanning tree on a weighted graph.*

Problem 21. *So far we have only looked at graphs with positively weighted edges. Will Dijkstra's algorithm and Prim's algorithm also work correctly on graphs with negatively weighted edges?*