Vocabulary 1

- **Graph**: A set of dots and connecting links
- **Vertex**: A dot on a graph (plural: vertices)
- **Edge**: A link on a graph
- **Path**: A connected sequence of edges showing a route on the graph that starts and ends at a vertex
- **Circuit**: A path that starts and ends at the *same* vertex

Vocabulary 2

- **Euler**: Swiss mathematician (pronounced *oiler*)
- **Euler circuit**: A circuit that covers each edge of a graph exactly once
- **Valence** of a vertex: The number of edges that meet at that vertex
- **Connected graph**: A graph is *connected* if for every pair of its vertices, there is at least one path connecting the two vertices

Euler

- **Euler's Theorem**
  - If a graph G is *connected* and has all *even* valences, then graph G has an *Euler circuit*.
  - If a graph G has an *Euler circuit*, then graph G must be *connected* and all its valences must be *even*.
- **Eulerizing**
  - If a graph G does not have an Euler circuit, we can add edges to it in order to make all valences even. This is called *Eulerizing* the graph.

Chinese Postman Problem

- Suppose a graph G does not have an Euler circuit. We can Eulerize this graph by duplicating existing edges.
- How can we choose which edges to re-use in order to minimize the total length of the graph?
- Simplified Chinese Postman Problem
  - Assume all edges have the same length.

Chinese Postman Problem

- **Algorithm**: A step-by-step procedure for solving a problem
- Develop an algorithm for finding some solution to the Simplified Chinese Postman Problem
  - It need not be the best possible solution
  - It just needs to produce an Eulerized graph

Vocabulary 3

- **Hamiltonian circuit**: A circuit that visits each vertex of a graph exactly once
  - Hamilton: 19th-century Irish mathematician
- Application: The traveling salesman problem
- Procedures for finding a Hamiltonian circuit:
  - Brute force
  - Sorted edges
  - Nearest neighbor

Spanning Trees

- A **spanning tree** is a graph containing *no circuits* that connects every vertex to every other vertex
- **Kruskal's algorithm** gives the minimum cost spanning tree
  - List the edges in order of length (cost)
  - Go through the list from smallest cost to largest
  - Activate the edge only if it does not form a circuit
  - Stop when all vertices are connected
- Applications: Building roads, phone networks