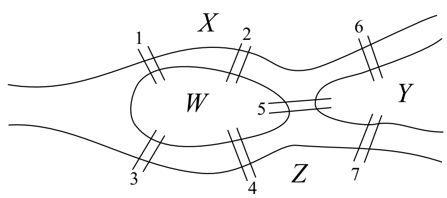
Introduction to Graph Theory

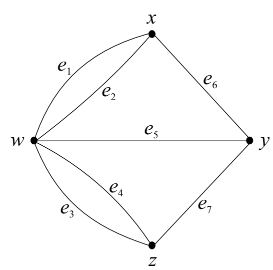
Allen Dickson

October 2006

1 The Königsberg Bridge Problem

The city of Königsberg was located on the Pregel river in Prussia. The river divided the city into four separate landmasses, including the island of Kneiphopf. These four regions were linked by seven bridges as shown in the diagram. Residents of the city wondered if it were possible to leave home, cross each of the seven bridges exactly once, and return home. The Swiss mathematician Leonhard Euler (1707-1783) thought about this problem and the method he used to solve it is considered by many to be the birth of graph theory.





Exercise 1.1. See if you can find a round trip through the city crossing each bridge exactly once, or try to explain why such a trip is not possible.

The key to Euler's solution was in a very simple abstraction of the puzzle. Let us redraw our diagram of the city of Königsberg by representing each of the land masses as a vertex and representing each bridge as an edge connecting the vertices corresponding to the land masses. We now have a graph that encodes the necessary information. The problem reduces to finding a "closed walk" in the graph which traverses each edge exactly once, this is called an Eulerian circuit. Does such a circuit exist?

2 Fundamental Definitions

We will make the ideas of graphs and circuits from the Königsberg Bridge problem more precise by providing rigorous mathematical definitions.

A **graph** G is a triple consisting of a **vertex set** $V(G)$, an **edge set** $E(G)$, and a relation that associates with each edge, two vertices called its **endpoints** (not necessarily distinct).

Graphically, we represent a graph by drawing a point for each vertex and representing each edge by a curve joining its endpoints.

For our purposes all graphs will be **finite graphs**, i.e. graphs for which $V(G)$ and $E(G)$ are finite sets, unless specifically stated otherwise.

Note that in our definition, we do not exclude the possibility that the two endpoints of an edge are the same vertex. This is called a **loop**, for obvious reasons. Also, we may have **multiple edges**, which is when more than one edge shares the same set of endpoints, i.e. the edges of the graph are not uniquely determined by their endpoints.

A **simple graph** is a graph having no loops or multiple edges. In this case, each edge e in $E(G)$ can be specified by its endpoints u, v in $V(G)$. Sometimes we write $e = uv$.

When two vertices u, v in $V(G)$ are endpoints of an edge, we say u and v are **adjacent**.

A **path** is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the ordering. A path which begins at vertex u and ends at vertex v is called a u, v-path.

A **cycle** is a simple graph whose vertices can be cyclically ordered so that two vertices are adjacent if and only if they are consecutive in the cyclic ordering.

We usually think of paths and cycles as subgraphs within some larger graph.

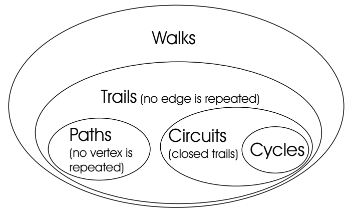
A **subgraph** H of a graph G, is a graph such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ satisfying the property that for every $e \in E(H)$, where e has endpoints $u, v \in V(G)$ in the graph G, then $u, v \in V(H)$ and e has endpoints u, v in H, i.e. the edge relation in H is the same as in G.

A graph G is **connected** if for every $u, v \in V(G)$ there exists a u, v-path in G. Otherwise G is called **disconnected**. The maximal connected subgraphs of G are called its **components**.

A **walk** is a list $v_0, e_1, v_1, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i. A **trail** is a walk with no repeated edge. A u, v-walk or u, v-trail has first vertex u and last vertex v. When the first and last vertex of a walk or trail are the same, we say that they are closed. A closed trail is called a **circuit**.

With this new terminology, we can consider paths and cycles not just as subgraphs, but also as ordered lists of vertices and edges. From this point of view, a path is a trail with no repeated vertex, and a cycle is a closed trail (circuit) with no repeated vertex other than the first vertex equals the last vertex.

Alternatively, we could consider the subgraph traced out by a walk or trail.

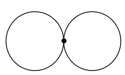
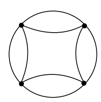
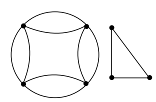
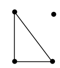


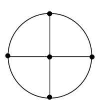
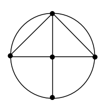
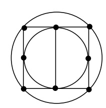
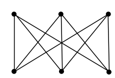
An **Eulerian trail** is a trail in the graph which contains all of the edges of the graph. An **Eulerian circuit** is a circuit in the graph which contains all of the edges of the graph. A graph is **Eulerian** if it has an Eulerian circuit.

The **degree** of a vertex v in a graph G, denoted $\deg v$, is the number of edges in G which have v as an endpoint.

3 Exercises

Consider the following collection of graphs:

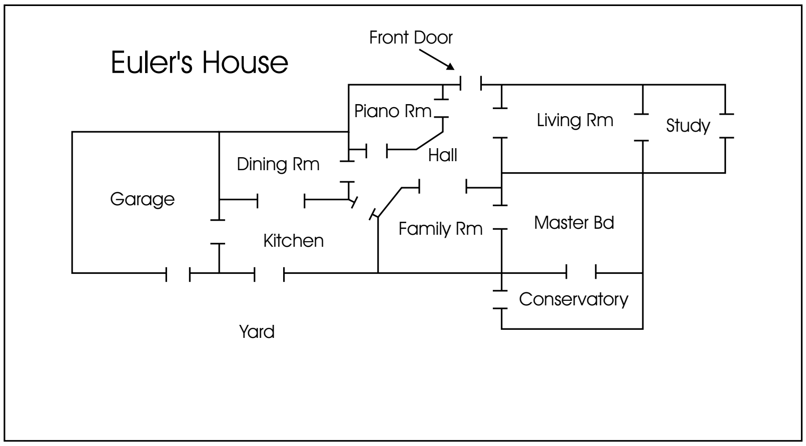
(a)  (b)  (c)  (d) 

(e)  (f)  (g)  (h) 

1. Which graphs are simple?
2. Suppose that for any graph, we decide to add a loop to one of the vertices. Does this affect whether or not the graph is Eulerian?
3. Which graphs are connected?
4. Which graphs are Eulerian? Trace out an Eulerian circuit or explain why an Eulerian circuit is not possible.
5. Are there any graphs above that are not Eulerian, but have an Eulerian trail?
6. Give necessary conditions for a graph to be Eulerian.
7. Give necessary conditions for a graph to have an Eulerian trail.

8. Given that a graph has an Eulerian circuit beginning and ending at a vertex v, is it possible to construct an Eulerian circuit beginning and ending at any vertex in the graph?

9. Euler's House. Baby Euler has just learned to walk. He is curious to know if he can walk through every doorway in his house exactly once, and return to the room he started in. Will baby Euler succeed? Can baby Euler walk through every door exactly once and return to a different place than where he started? What if the front door is closed?



4 Characterization of Eulerian Circuits

We have seen that there are two obvious necessary conditions for a graph to be Eulerian: the graph must have at most one nontrivial component, and every vertex in the graph must have even degree. Now a more interesting question is, are these conditions sufficient? That is, does every connected graph with vertices of even degree have an Eulerian circuit? This is the more difficult question which Euler was able to prove in the affirmative.

Theorem 1. *A graph G is Eulerian if and only if it has at most one nontrivial component and its vertices all have even degree.*

There are at least three different approaches to the proof of this theorem. We will use a constructive proof that provides the most insight to the problem. There is also a nonconstructive proof using maximality, and a proof that implements an algorithm.

We will need the following result.

Lemma 2. *If every vertex of a graph G has degree at least 2, then G contains a cycle.*

Proof. Let P be a maximal path in G. Maximal means that the path P cannot be extended to form a larger path. Why does such a path exist? Now let u be an endpoint of P. Since P is maximal (cannot be extended), every vertex adjacent to u must already be in P. Since u has degree at least two, there is an edge e extending from u to some other vertex v in P, where e is not in P. The edge e together with the section of P from u to v completes a cycle in G. ■

Proof of theorem. We have already seen that if G is Eulerian, then G has at most one nontrivial component and all of the vertices of G have even degree. We just need to prove the converse.

Suppose G has at most one nontrivial component and that all of the vertices of G have even degree. We will use induction on the number of edges n.

Basis step: When $n = 0$, a circuit consisting of just one vertex is an Eulerian circuit.

Induction step: When $n > 0$, each vertex in the nontrivial component of G has degree at least 2. Why? By the lemma, there is a cycle C in the nontrivial component of G. Let G' be the graph obtained from G by deleting $E(C)$. Note that G' is a subgraph of G which also has the property that all of its vertices have even degree. Why? Note also that G' may have several nontrivial components. Each of these components of G' must have number of edges $< n$. Why? By the induction hypothesis, each of these components has an Eulerian circuit. To construct an Eulerian circuit for G, we traverse the cycle C, but when a component of G' is entered for the first time (why must every component intersect C?), we detour along an Eulerian circuit of that component. This circuit ends at the vertex where we began the detour. When we complete the traversal of C, we have completed the Eulerian circuit of G. ■