COMP 251: Data Structures and Algorithms Winter 2009

Eulerian circuits

Prepared by: Ethan Kim

The Problem of the Königsberg Bridges

In the old city of Königsberg in Eastern Prussia (now renamed Kaliningrad), there is a river flowing through the city, and an island called Kneiphof. There are 7 bridges connecting different regions of the land in this area, and people of Königsberg used to wonder if it is possible to find a tour that would cross each of the seven bridges exactly once. A simplified map of this region is shown in Figure 1.

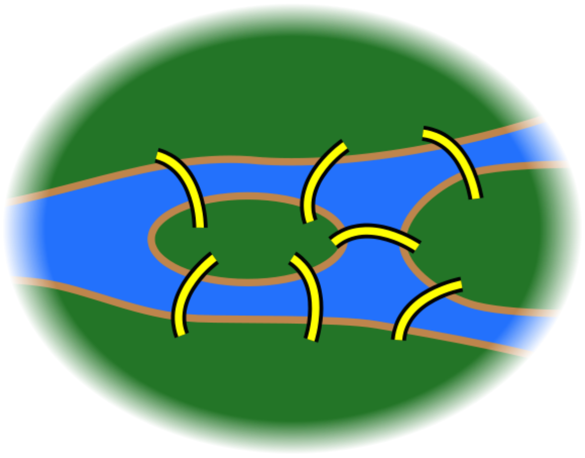
Figure 1: A simplified map of the 7 bridges in Königsberg.

Since nobody was able to devise such a tour, many believed that this is impossible. However, this "puzzle" hadn't been treated mathematically until 1736, when Leonhard Euler wrote an article which dealt with this problem, and gave a general method for other problems of the same type. In his 1736 article, Euler treated the lands and bridges as abstract objects, which are later used to develop the theory of graphs.

Graph Theory Terminology

Here is a (short) list of terminology in graph theory. Make sure you know them.

- vertex(node), edge
- multigraph vs. simple graph
- degree (indegree, outdegree)

- path, cycle
- walk, circuit
- connected, connected component
- , and so on..

Eulerian Circuits

A graph is said to contain an *Eulerian circuit*, if there exists a circuit that visits every edge precisely once. The following result was given in Euler's 1736 paper:

Theorem 1. *A graph G contains an Eulerian circuit if and only if the degree of each vertex is even.*

The proof of necessity is easy to see. Suppose G contains an Eulerian circuit C. Then, for any choice of vertex v, C contains all the edges that are adjacent to v. Furthermore, as we traverse along C, we must enter and leave v the same number of times, and it follows that $deg(v)$ must be even.

While this proof of necessity was given by Euler, the proof of converse is not stated in his paper. It is not until 1873 (137 years later) when a young German mathematician, Carl Hierholzer published the proof of sufficiency[1].

Proof of Sufficiency. We prove by induction on the number of edges. For graphs with all vertices of even degree, the smallest possible number of edges is 3 (i.e. a triangle) in the case of simple graphs, and 2 (i.e. a digon) in the case of multigraphs. In both cases, the graph trivially contains an Eulerian circuit. The induction hypothesis then says:

Let H be a connected graph with $\leq k$ edges. If every vertex of H has even degree, H contains an Eulerian circuit.

Now, let G be a graph with $k+1$ edges, and every vertex has an even degree. Since there is no odd degree vertex, G cannot be a tree. Thus, G must contain a cycle C. Now, remove the edges of C from G, and consider the remaining graph G'. Since removing C from G may disconnect the graph, G' is a collection of connected components, namely $G'_1, G'_2, \ldots$, etc. Furthermore, when the edges in C are removed from G, each vertex loses even number of adjacent edges. Thus, the parity of each vertex is unchanged in G'. It follows that, for each connected component of G', every vertex has an even degree. Therefore, by the induction hypothesis, each of $G'_1, G'_2, \ldots$ has its own Eulerian circuit, namely C_1, C_2, etc.

We can now build an Eulerian circuit for G. Pick an arbitrary vertex a from C. Traverse along C until we reach a vertex v_i that belongs to one of the connected components G'_i. Then, traverse along its Eulerian circuit C_i until we traverse all the edges of C_i. We are now back at v_i, and so we can continue on along C. In the end, we shall return back to the first starting vertex a, after visiting every edge exactly once. □

[1]In fact, this 1873 paper was prepared by C. Wiener, a colleague of Hierholzer, due to Hierholzer's early death. Neither Wiener nor Hierholzer was aware of Euler's earlier work (as may be explained by difficulties of communication at the time), and Hierholzer's work involved systems of intersecting lines on the plane.

The theorem, however, is only an existential statement. If the necessary and sufficient condition is satisfied, one would wish to find an Eulerian circuit. It is easy to see that the above inductive proof naturally gives an algorithm to construct Eulerian circuits – recursively find a cycle, and then remove the edges of the cycle. Here we study another algorithm that finds Eulerian circuits.

Fleury's algorithm, as shown below, is based on a simple observation: if we can traverse all the edges while removing them in such a way that all the remaining edges stay connected, we would obtain an Eulerian circuit. This can be realized by making sure we never pick a cut edge(bridge) as the next edge to traverse[2]. Hence, we have the following algorithm.

Fleury's Algorithm

1. Check if the graph is connected, and every vertex is of even degree. Reject otherwise.
2. Pick any vertex v_{start} to start.
3. While the graph contains at least one edge:
   (a) Pick an edge that is not a bridge.
   (b) Traverse that edge, and remove it from G.

Proof of Correctness. To see that this algorithm terminates, suppose we are stuck at some vertex v before all the edges are removed. Then either v must have an odd degree, or G is not a connected graph to begin with. Both of these cases are impossible due to our first check. This is a contradiction.

Note that, when the algorithm terminates, we must return to v_{start} because every vertex has an even degree. Furthermore, the edge removal operation guanrantees that each edge is visited exactly once. Therefore, the discovered tour is an Eulerian circuit. $\square$

Variation 1. Eulerian Paths

One may also ask about a walk, not necessarily closed, that visits each edge precisely once[3]. Since the number of odd degree vertices was used to characterize Eulerian circuits, a nice approach would be to consider this number. In his original paper, Euler pointed out another classical result in graph theory.

Handshaking Lemma. [4] *Every graph has even number of odd degree vertices.*

Proof. Consider the sum of the degree of all vertices, $S = \sum_{u \in V} \deg(u)$. In this sum, every edge (a, b) in the graph gets counted twice: once for a, and once for b. Therefore, $S = 2m$ is an even number. Now, let V_{odd} (V_{even}) denote the subset of V that consists only of odd (even,

[2]Here, as we remove edges from the graph, we also remove any vertex that has no adjacent edges. In other words, it is okay to remove an edge if it makes a leaf node into an isolated vertex.

[3]In fact, this is the version that was discussed in Euler's original paper.

[4]This name comes from the following question: if a collection of guests at a party shake hands when they meet, how many people shook hands with odd number of people?

respectively) degree vertices. Since $S = \sum_{u \in V} \deg(u) = \sum_{v \in V_{even}} \deg(v) + \sum_{w \in V_{odd}} \deg(w)$, the number $\sum_{w \in V_{odd}} \deg(w)$ must be even. Therefore, $|V_{odd}|$ must be even. $\square$

By this result, we know the possible number of odd degree vertices in a graph is 0, 2, 4, etc. When there is no odd degree vertex, as we have seen above, there is an Eulerian circuit. When there are two odd degree vertices, we would have an Eulerian path with the two odd degree vertices serving as initial and final vertex, respectively.

Theorem 2. *A graph contains an Eulerian path if and only if there are 0 or 2 odd degree vertices.*

Proof. Suppose a graph G contains an Eulerian path P. Then, for every vertex v, P must enter and leave v the same number of times, *except* when it is either the starting vertex or the final vertex of P. When the starting and final vertices are distinct, there are precisely 2 odd degree vertices. When these two vertices coincide, there is no odd degree vertex.

Conversely, suppose G contains 2 odd degree vertex u and v. (The case where G has no odd degree vertex is shown in Theorem 1.) Then, temporarily add a dummy edge (u, v) to G. Now the modified graph contains no odd degree vertex. By Theorem 1, this graph contains an Eulerian circuit C that also contains (u, v). Remove (u, v) from C, and now we have an Eulerian path where u and v serve as initial and final vertices. $\square$

This trick of adding a dummy edge can also be applied to modify Fleury's algorithm. Note that it is okay to add a dummy edge (u, v) even if the original graph already contains the edge (u, v) – Theorem 1 holds for multigraphs as well, and the original Königsberg graph was also a multigraph.

Exercise 1. Euler circuits in Complete Graphs Given n points in the plane, we would like to find a circuit that draws line segments between *every* pair of points, and we want that circuit to be Eulerian. In other words, we would like to find Eulerian circuits for a complete graph K_n. Is this possible? If so, which complete graphs admit an Eulerian circuit?

Variation 2. Eulerian circuit/path in Directed Graphs

This section is added after the lecture. What happens when the graph is directed? It is easy to see that, similar arguments apply if we consider indegree vs. outdegree of each vertex[5].

Theorem 3. *Let $D = (V, A)$ be a directed graph. Then D contains an Eulerian* circuit *if and only if, for every vertex $u \in V$, $indeg(u) = outdeg(v)$.*

Furthermore, D contains an Eulerian path *if and only if, there exists two vertices s and t such that:*

$$\begin{aligned} outdeg(s) &= indeg(s) + 1 \\ indeg(t) &= outdeg(t) + 1 \\ indeg(v) &= outdeg(v), \qquad \forall v \in V - \{s, t\} \end{aligned}$$

[5]Indegree of a vertex v is the number of edges that points to v, whereas outdegree of v is the number of edges that leave from v.

Proof. Try it! □

Hamiltonian Circuits

We can change our problem slightly: if a closed circuit visits every *vertex* exactly once, it is called a *Hamiltonian circuit.* Similarly, if there is a path, not necessarily closed, that visits every vertex exactly once, it is called a *Hamiltonian path.* We say a graph is *Hamiltonian* if it contains a Hamiltonian circuit. While these problems seem strikingly similar to the Eulerian counterparts, no efficient solution is known. As shown in Figure 2, these two classes of graphs are not quite related:

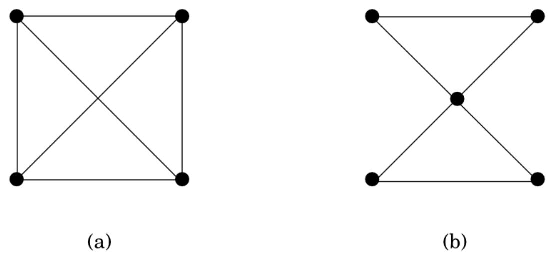

Figure 2: Hamiltonian vs. Eulerian: (a) This graph is Hamiltonian, but not Eulerian. (b) This graph is Eulerian, but not Hamiltonian.

- Not all Hamiltonian graphs are Eulerian.
- Not all Eulerian graphs are Hamiltonian.

Applications of Eulerian Paths

In Biology, researchers wish to discover the DNA sequence of the human Genome. There has been an enormous amount of effort to read the whole human DNA(which is approximately 2 billion in length), but the current technology allows us to read only a short fragments ($\approx$ 20 characters). So, instead of trying to read the whole DNA sequence, they break the DNA sequences into shorter fragments, and then would read each fragment one by one. This is called shotgun sequencing.

From these short fragments, they wish to reconstruct the entire sequence. If these fragments come from a single copy of a sequence, then clearly it is impossible to reconstruct. However, if the fragments come from many copies (in fact, millions) of the same DNA sequence, and each copy is broken into fragments at random positions, then we may be able to do the reconstruction.

Suppose we are given a collection of fragments, each of length 20. Assuming that these fragments cover *every* 20 character substring of the DNA, we can construct a directed graph as follows:

- $V = \{$ set of all 19 charcter substring from the fragments$\}$

- $A = \{$ set of all the fragments$\}$

Edited: Observe that when this graph is constructed, the edges now have directions associated with the fragments. For example, the fragment $ACTG$ would correspond to a directed edge going from a vertex ACT to a vertex CTG.

In other words, the graph we construct has all the fragments as edges, and the two vertices at the endpoints of an edge correspond to the overlap between the current fragment to the next. Then, if this graph contains an Eulerian path, then it would be a good prediction of the entire DNA sequence[6].

Of course, if the fragment lengths are so short that each fragment appears multiple times within the DNA sequence (which is often the case), then we cannot assume that each edge must be traversed exactly once – the whole DNA sequence may be traversing each edge more than once. This gives rise to another problem of similar taste, called *Chinese Postman problem.*

Exercise 2. Eulerization This problem is a bit harder, but nice to think about nonetheless. Suppose we are given a graph that contain neither an Eulerian circuit, nor an Eulerian path. If we allow multiple visits to each edge, can you find a circuit(or a path) that visits each edge *at least* once? If so, can you find a shortest such a tour (measured by the total length of the circuit or path)? This problem is called *Eulerization*, or *Chinese Postman Tour*: "Given a graph G, find the shortest circuit (or path) that visits each edge at least once."

[6] An earlier attempt was to construct the graph so that each fragment corresponds to a *vertex*, and two vertices are joined by an edge if the two fragments overlap. Then the entire DNA sequence would correspond to a Hamiltonian path of the graph – which is a difficult problem to solve.