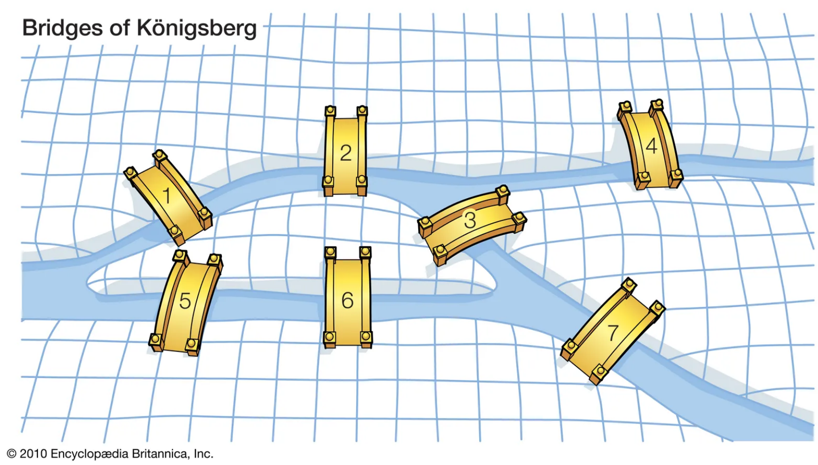
PROOF FOR CLOSED EULERIAN TRAILS

GRAHAM SWAIN

Abstract. In this paper we explore a proof showing that a graph contains a closed Eulerian trail if and only if all of its vertices have even degree. We show this with a direct proof and by developing an algorithm.

1. Introduction

This problem dates back to 18th century Königsberg, Prussia, modern day Kaliningrad, Russia. Königsberg had seven bridges that connected four landmasses, the figure below shows roughly how the bridges were arranged. The citizens of Königsberg wonder if a person could walk across town and cross each bridge exactly once.



The problem was solved by Leonard Euler, who found it to be impossible. His reasoning was that every time you enter a landmass you must also be able to leave it, so any landmass with an odd number of bridges coming from it would cause you to get stuck. There is an exception for the landmass that you start and as well as the one you end on, but in Königsberg each landmass has an odd number of bridges coming from them. This led to the theorem we are looking at today as well as set the basis for modern day graph theory.

2. Definitions

Before we look at the theorem, we first must go over some definitions to help understand it.

Definition *A **graph** $G = (V,E)$ is comprised of a set of vertices V and a set of different unordered pairs of distinct vertices from V called E. Elements from E are called edges.*

Definition *Two vertices v, w that exist in V are said to be **adjacent** if the edge vw exists in E.*

Definition *If ab exists in E, then we say that the vertex v and the edge vw are **incident**; w would also be incident with vw.*

Definition *The **degree** of a vertex v that exists in V is the amount of edges in E that are incident with v.*

Definition *A **walk** W is a finite sequence of vertices in which each consecutive pair of vertices are adjacent.* Vertices and edges are allowed to be repeated in a walk, with the exception of the same vertex consecutively, to prevent loops.

Definition *A **trail** T is a walk in which edges are not allowed to be repeated. A trail is called a **closed trail** when the trail begins and ends at the same vertex.*

Definition *A trail that contains all of the edges in E is called an **Eulerian trail**. A **closed Eulerian trail** when the trail begins and ends at the same vertex.*

Definition *A graph is said to be **connected** if, and only if, there is a walk between any two vertices v and w.*

3. Theorem and Proof

Theorem 1. *A connected graph has a closed Eulerian trail if and only if all of its vertices have even degree.*

Proof. ($\Rightarrow$) Assume a connected graph G has a closed Eulerian trail, denoted as C. Prove that every vertex has an even degree.

Take an arbitrary vertex v. As we traverse C, each time we enter v we must be able to exit on a distinct edge. Thus every vertex must be incident with $2k$ edges, where k is the number of times a vertex is visited. Therefore every vertex has an even degree.

The first vertex of C does make a special case. It does not matter since we chose an arbitrary vertex in the first step, but we can still look at it. Denote the first vertex of C as a. We know that a is incident to the first edge of C, the last edge of C, as well as a $2k$ amount of other edges. So a is adjacent to

$$1 + 2k + 1 = 2 + 2k$$

edges, which will result in an even number.

($\Leftarrow$) Assume G is a connected graph with every vertex having even degree. Prove that it has a closed Eulerian trail.

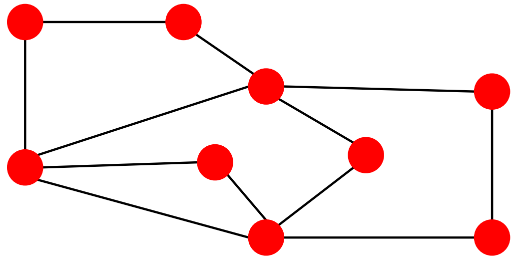


FIGURE 1. Graph G we will look at as we construct the algorithm.

To prove this direction we must construct an algorithm.

Step 1. Pick an arbitrary vertex v to act as our starting point.

Step 2. Construct a trail beginning at v. Continue to add edges to this trail until we arrive back at v. (*We know we arrived at v since if we entered another vertex a that we could not leave then a has an edge to enter on but not a corresponding edge to exit on. Which would mean that a has an odd degree, which would be a contradiction.*) Since we did arrive back at v we formed a closed trail, which we will called C_1.

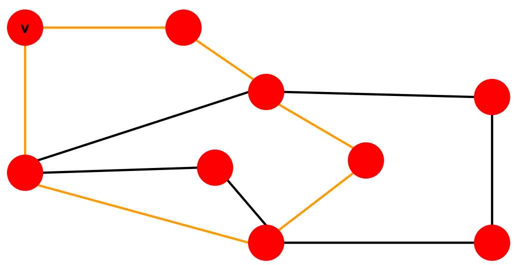


FIGURE 2. C_1 is highlighted in orange.

Step 3. Check if C_1 contains all the edges of G. If it does we have found a closed Eulerian trail and are done. If it does not contain all of the edges of G then we continue to **Step 4**.

Step 4. Remove all of the edges in C_1 from G. We will call the resulting graph G'. (G' does not need to be connected).

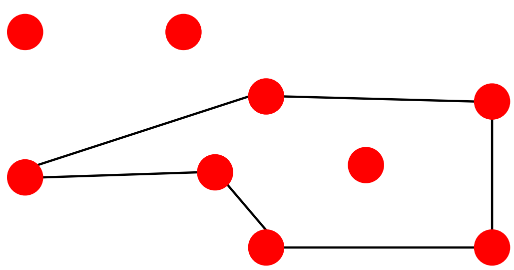


FIGURE 3. The remaining edges after we remove C_1 from G form G'.

Step 5. Choose an arbitrary vertex w that is in both C_1 and G.

Step 6. Construct a trail in G' beginning at w and continue until we arrive back at w. We will call the resulting closed trail C_2.

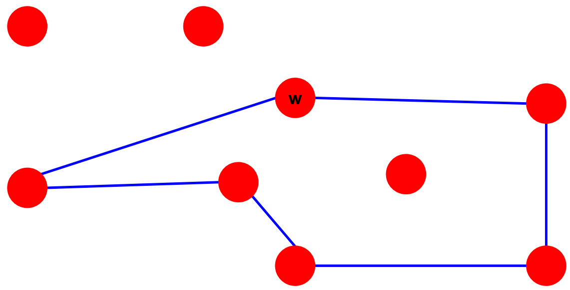


FIGURE 4. C_2 is highlighted in blue.

Step 7. Combine C_1 and C_2 to form a new closed trail denoted as C'. If C' contains all of the edges of G, we have found a closed Eulerian trail and are done. If not continue to **Step 8.**

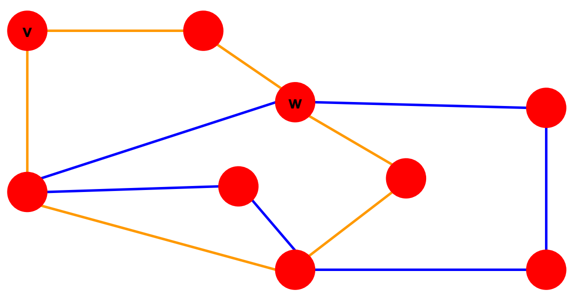


FIGURE 5. C' is formed by combining C_1 and C_2. C' does contain all of the edges of G, so we are done in this example.

We know that combining two closed trails form a new closed trail. To show this, start at v in C_1. Trace C_1 until we arrive at w. Once we reach w, trace the entirety of C_2. Once we arrive back at w, continue tracing C_1 until we end at v. The trace is C'.

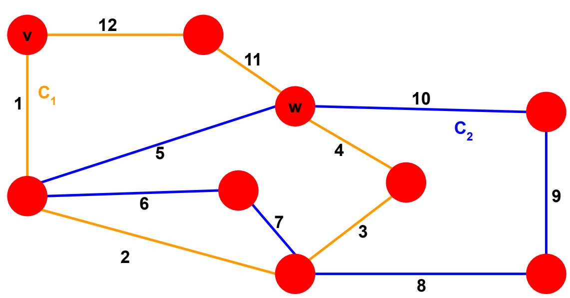


FIGURE 6. Followed the number trail to see how two closed trails can be combined to formed another closed trail.

Step 8. Remove all of the edges of C' from G to get G''. Chose a vertex in both C' and G'' and continue in this manner until we have removed all the edges of G.

□

This theorem is highly related to another theorem dealing with Eulerian trails.

Theorem 2. *A connected graph has an Eulerian trail but not an closed Eulerian trail if and only if it has exactly two vertices of odd degree.*

We will not make a formal proof of this theorem, but we can look at it intuitively. The difference between an Eulerian trail and closed Eulerian trail is that a closed Eulerian trail begins and ends at the same vertex. So to have an Eulerian trail that is not closed we know that the first vertex a must be distinct from the last vertex z. In the proof of Theorem 1 we found that the beginning vertex must be incident to $2k+2$ edges, since the first edge and the last edge are both incident to it. So it follows that a and z would be incident to $2k+1$ edges, since a is incident to the first edge but not the last and the opposite is true for z. All of the other edges would still be incident to $2k$ edges. For the other direction we can prove using the same algorithm. The only difference is we need to make a new "fake" edge between a and z to temporarily form a closed Eulerian trail. Once we are done with the algorithm we can remove az to return to a Eulerian trail that is not closed.

4. Conclusion

It should now be clear why it was impossible to take a trip across Königsberg and cross each bridge exactly once. Figure 7 shows the bridges as a graph, with the landmass as the vertices and the bridges as the edges. You can see how the vertex on the left has a degree of five and that the others have degree three, so there is no closed Eulerian trail.

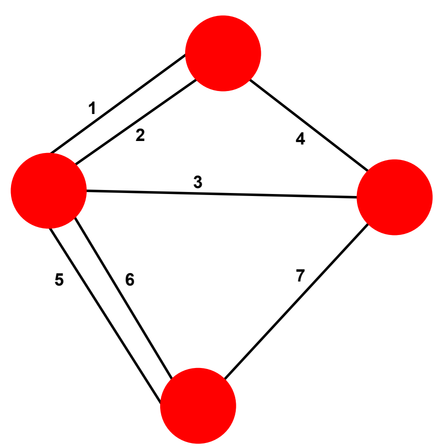


Figure 7. A graphical representation of the bridges of Königsberg.

The bridges of Königsberg is also a great example of how we can use graph theory to take information from the real world and simplify it to make it easier to study. It is for this reason that Eulerian trails have many practical uses. The most common use of them is likely navigation, especially for companies like mail carriers. These companies can look at roads as edges and houses as vertices, and they need to visit every vertex as efficiently as possible. So they would find a closed Eulerian trail (if one exists) beginning and terminating at their distribution center. Another use would be computer networking, for similar reasons. You want to be able to connect computers and servers as efficiently as possible. Closed Eulerian trails are also being used to help reconstruct DNA fragments. Eulerian trails and graph theory are often really useful when it comes to studying how complex systems are connected.

References

1. S. Carlson, Königsberg bridge problem: https://www.britannica.com/science/Konigsberg-bridge-problem
2. S. Belcastro, *Discrete Mathematics with Ducks*, CRC Press, 2012.
3. S. Epp, *Discrete Mathematics with Applications*, Brooks Cole, 1996.