Euler Circuits

INTRODUCTION
Euler wrote the first paper on graph theory. It was a study and proof that it was impossible to cross the seven bridges of Königsberg once and only once. Thus, an Euler Trail, also known as an Euler Circuit or an Euler Tour, is a nonempty connected graph that traverses each edge exactly once.

PROOF AND ALGORITHM
The theorem is formally stated as: “A nonempty connected graph is Eulerian if and only if it has no vertices of odd degree.” The proof of this theorem also gives an algorithm for finding an Euler Circuit.

- Let $G$ be Eulerian, and let $C$ be an Euler tour of $G$ with origin and terminus $u$. Each time a vertex $v$ occurs as an internal vertex of $C$, two of the edges incident with $v$ are accounted for. Since an Euler tour contains every edge of $G$, $d(v)$ is even for all $v \neq u$. Similarly, since $C$ starts and ends at $u$, $d(u)$ is also even. Thus $G$ has no vertices of odd degree.

EXAMPLE
I will use figure 1 as an example.

**Figure 1**

(i) Start at an arbitrary vertex $u$. Trace out a route $C$ that never repeats an edge. Since there are no vertices of an odd degree, each vertex that is arrived at can also be left by a different edge.
I created an arbitrary route $C$ in figure 2 that goes $A - D - E - F - A$.

**Figure 2**
(ii) If the route does not contain all edges, choose a vertex \( u' \) on \( C \) that is incident to an unused edge. Repeat step i) starting at \( u' \) and using the graph consisting only of the unused edges. Incorporate the resulting circuit into the original route \( C \).

As we can see in figure 2, the edges \( AB, BC, \) and \( AC \) are outside of our path \( C \). Vertex \( A \) is incident to two of these edges. We can trace out a new route \( D \) that consists of \( A \rightarrow B \rightarrow C \rightarrow A \).

We can then combine path \( C \) and \( D \), since we know there are no overlapping edges, to have a complete Euler Path that starts and ends at vertex \( A \), as seen in figure 3.

![Figure 3](image)

(iii) Repeat step ii) until all edges are included in route \( C \).

Our path is complete.

**BRIDGES OF KÖNIGSBERG**

Since the theorem has been proved, we can see the reason that the bridges of Königsberg, figure 4, cannot be a Euler Tour is that all the vertices have an odd degree: \( d(A) = 3, d(B) = 5, d(C) = 3, d(D) = 3 \) (see figure 5). But even if just one vertex had an odd degree, there could still not be an Euler Tour.

![Figure 4](image)

**PRACTICAL APPLICATIONS**

There are many applications of the Euler Circuit. Often times, it would seem that a Hamiltonian Cycle would be more useful, that the vertices would represent the locations we
would want to visit once and only once. But consider the scenario of a street sweeper. We want to clean each street, but we do not want to waste time and money by going over the same section of street more than once. Consider the situation of a typical grid-style city street design, as seen in *figure 6*.

![Figure 6](image)

Each edge represents a road. Six of the vertices have an odd degree. This, it is not possible to traverse each road once and only once. But consider the fact that each road has two sides, and a street sweeper must sweep both sides of the street (see *figure 7*). Now, each vertex has twice as many edges connected to it since each edge actually represents two passes that must be made by the street sweeper. Each vertex previous had degree of 2, 3, or 4. Now, since the degree of each vertex has doubled, each vertex has a degree of 4, 6, or 8. All of these are even numbers, which means that all vertices have an even degree. This means that an Euler Circuit should be possible.

![Figure 7](image)

Here is another example. Consider a paper-delivery boy. He can throw papers to either side of the road as he bikes up the street. He has to cover every street but does not want to repeat a street since that would take extra time. Considering *figure 8*, would the paper boy be able to bike in an Euler Circuit?

![Figure 8](image)
Yes, because each vertex has an even degree. To find the Euler Circuit using the above algorithm, we would have to repeat the steps three times.

**AN ALTERNATE ALGORITHM**

Alan Tucker presents a new proof of Euler’s Theorem:

- At each vertex arbitrarily pair off, and link together, the incident edges. The result is a set of chains that, having no ends, must be circuits.
- Repeatedly combine any two circuits with a common vertex until we have vertex-disjoint circuits.
- Since the graph is connected, there is just one circuit – an Euler Circuit.

From this theorem, I devised a new algorithm. For each vertex, list all incident edge pairs. I will use figure 8 as an example. My example is more simplistic in that direction does not matter. We simply do not want to repeat an edge, but also must not miss an edge. Vertex 1 has one two-edge chain: 5 – 1 – 2. Vertex 2 also has only one two-edge chain: 1 – 2 – 3. I can continue making these two-edge chains for all vertices:

1. 5 – 1 – 2
2. 1 – 2 – 3
3. 2 – 3 – 7
4. 8 – 4 – 5
5. 4 – 5 – 6, 9 – 5 – 6, 1 – 5 – 6, 1 – 5 – 4, 4 – 5 – 9
6. 5 – 6 – 10, 5 – 6 – 7, 10 – 6 – 7, 10 – 6 – 11, 5 – 6 – 11
7. 6 – 7 – 11, 6 – 7 – 10, 10 – 7 – 11, 3 – 7 – 11, 6 – 7 – 3, 10 – 7 – 3
8. 4 – 8 – 9
9. 8 – 9 – 5
10. 6 – 10 – 7
11. 6 – 11 – 7

To create our circuit, we first find the vertices that have only one two-edge pair. These are indicated in bold. We must use each of these edge paths. We will pick an arbitrary starting path. I choose 5 – 1 – 2. Now we must find another path that starts or ends at 2. I choose 2 – 3 – 7. Next, I choose 7 – 11 – 6, then 6 – 10 – 7. As I progress, I must eliminate any two-path circuit that contains these paths. I must eliminate any path that has edges 5 – 1, 1 – 2, 2 – 3, 3 – 7, 7 – 11, 11 – 6, 6 – 10, and 10 – 7. The new set of remaining edges is:

1.
2.
3.
4. 8 – 4 – 5
5. 4 – 5 – 6, 9 – 5 – 6, 4 – 5 – 9
6. 5 – 6 – 7
7.
8. 4 – 8 – 9
9. 8 – 9 – 5
10.
11.
We see that we now have only one option as a way to pass through the *vertex 6*. We can complete the rest of the circuit in this manner so that we have \(5 - 1 - 2 - 3 - 7 - 11 - 6 - 10 - 7 - 6 - 5 - 4 - 8 - 9 - 5\). We have traversed each edge without ever repeating one. Also, you can see that is clearly okay to pass through a vertex more than once, as 5 and 6 and 7 are repeated. I found this algorithm a very useful way to help me logically create an Euler Circuit.

**Necessity of an Algorithm**

But you might wonder why, with the availability of computing, we need an algorithm at all? Why not just use a brute force method that tries each possible directed graph, and check stop when one does form an Euler Circuit? Consider the number of possibilities in a simple triangle graph. There are three edges. Each edge can be directed in one of two directions. So there are \(2^3 = 8\) possibilities. Assume constructing and checking each possibility takes one second. Then finding the Euler Circuit in a triangle takes eight seconds. But what about using the brute force method for larger graphs, such as the one shown in *figure 7*? In that example, there are 34 edges. This means there are \(2^{34}\) combinations. Again assuming one second for each combination, it would take over 500 years to try every combination. That was not even a very large example we were considering. Obviously, an algorithm is necessary to find an Euler Circuit computationally.

**Hamiltonian Cycles and Euler Circuits**

Hamiltonian Cycles and Euler Circuits are similar. While an Euler Circuit requires that each edge be traversed once and only once, a Hamiltonian Cycle requires that each vertex be visited once and only once. Is it possible for a graph to have both a Hamiltonian Cycle and an Euler Circuit? First, we must know what conditions are necessary for a Hamiltonian Cycle.

- For all \(v\) belong to \(V\), \(\text{deg}(v) \geq 2\)
- If \(a\) belongs to \(V\) and \(\text{deg}(a) = 2\), then the two edges incident with vertex \(a\) must appear in every Hamilton cycle for \(G\)
- If \(a\) belongs to \(V\) and \(\text{deg}(a) > 2\), then as we try to build a Hamilton cycle, once we pass through a vertex \(a\), any unused edges incident with \(a\) are deleted from further consideration.

Let us start by considering a simple graph: the triangle. Each vertex has degree 2. Thus, we know both a Hamilton cycle and an Euler Circuit exist.

What is we consider \(K_5\), though? In this graph, there are \((n * (n – 1) / 2) = 10\) edges. To touch all 5 vertices only once, we can traverse at most 5 edges. This leaves 5 edges that are not traversed. Thus, even though there is a Hamiltonian Cycle, there cannot be an Euler Circuit.

From my considerations, it seems that a Hamiltonian Cycle and an Euler Circuit can coexist in a graph if and only if the number of vertices is equal to the number of edges, specifically when the degree of each vertex is two.
REFERENCES
