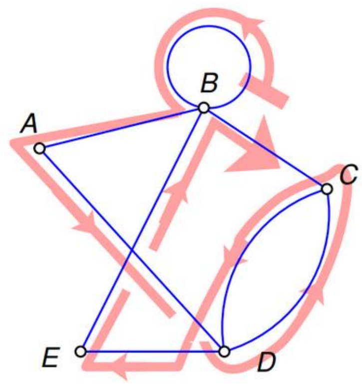
Euler Paths and Euler Circuits

An **Euler Path** is a path that goes through every *edge* of a graph exactly once

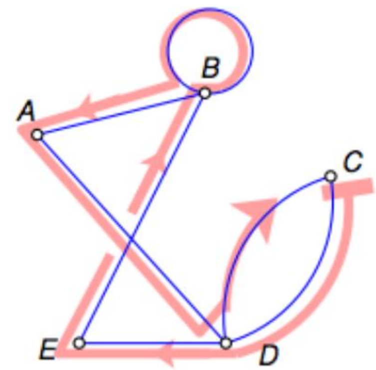
An **Euler Circuit** is an **Euler Path** that begins and ends at the same vertex.

Euler Path



Euler Path: BBADCDEBC

Euler Circuit



Euler Cicuit: CDEBADC

Euler's Theorem:

1. If a graph has more than **2** vertices of odd degree then it has no Euler paths.
2. If a graph is connected and has **0** or exactly **2** vertices of odd degree, then it has at least one Euler path
3. If a graph is connected and has **0** vertices of odd degree, then it has at least one Euler circuit.

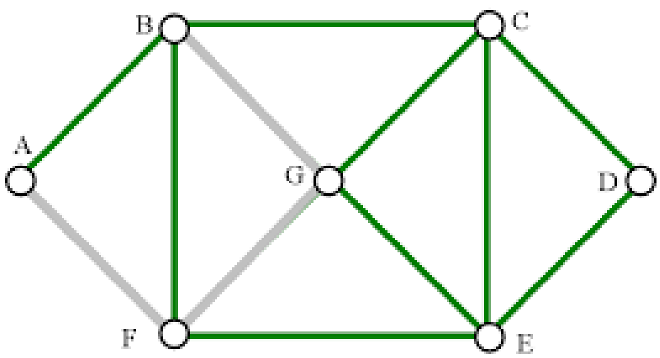
# Odd Vertices	Euler Path?	Euler Circuit?
0	***YES***	***YES***
2	***YES***	No
4, 6, 8 …	No	No
1, 3, 5 …	No Such Graphs Exist!!!	

Tracing a graph: A graph can be traced if you can begin at an edge and draw the entire graph without lifting up your pencil or going over an edge twice. If a graph contains two odd vertices, you must begin at one and end at the other.

Euler Paths and Euler Circuits

Finding an Euler Circuit: There are two different ways to find an Euler circuit.

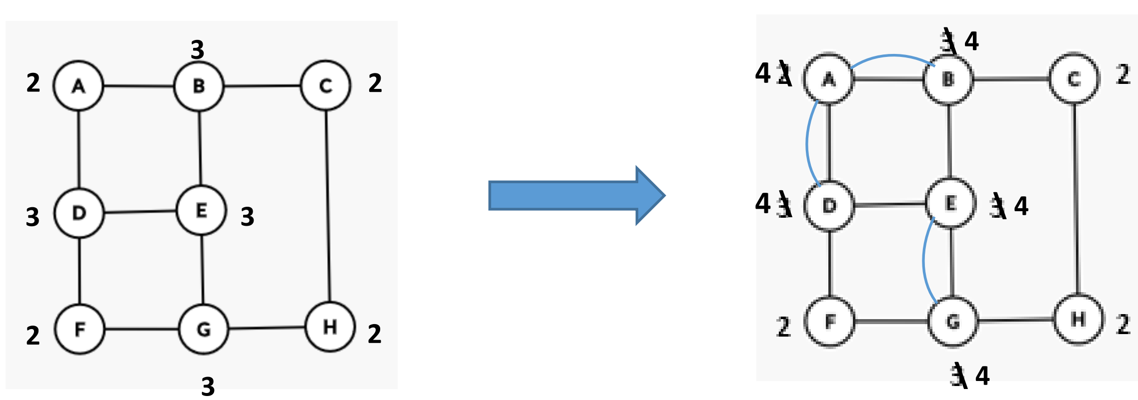
1. **Fleury's Algorithm:** Erasing edges in a graph with no odd vertices and keeping track of your progress to find an Euler Circuit.
   a. Begin at any vertex, since they are all even. A graph may have more than 1 circuit).
   b. After you have traveled over an edge, erase it. If all edges at a particular vertex have been erased, erase the vertex as well.
   c. Only travel over an edge that is a bridge if there is no other option.



Starting at vertex A you come through AFGB.
Then you cannot choose edge AB as it is a bridge.
Now BFECDEGCBA completes the trail.

2. **Eulerizing a Graph:** Repeating edges on a graph with odd vertices so that the graph has no odd vertices. *(Remember, there will always be an even number of odd vertices!)*

   a. Pick out all vertices of an odd degree.
   b. Repeat edges between vertices until the final graph has no odd vertices.
   c. **You must repeat pre-existing edges only!!!!**



*For this example, you can add edges **AB** and **AD**, but you **CANNOT** add **BD** because there isn't already an edge between vertices B and D.