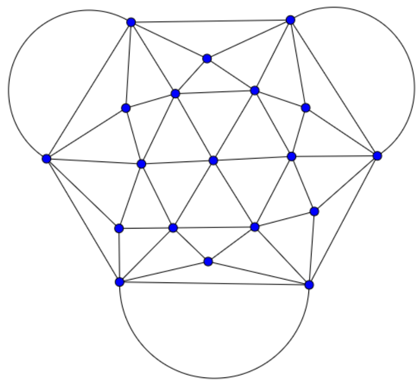
# Section 6.2 Edge Traversal Problems

**Definition:** A path is called an **Euler path** if it crosses every edge in a graph exactly once.
**Definition:** A closed path is called an **Euler circuit** if it crosses every edge in a graph exactly once.

**Theorem:** A graph that has an Euler circuit must have all vertices of even degree.
**Proof:**

**Corollary:** A graph that has an Euler path has either two vertices of odd degree or no vertices of odd degree.
**Proof:**

Does the graph below have an Euler Path and/or an Euler Circuit?



Be careful: the theorems we have so far just say that IF a graph has an Euler path it has at most two vertices of odd degree. It does NOT say the converse - IF a graph has 0 or 2 odd vertices then it has a Euler path.

To actually answer this question from what we know so far, you actually need to construct an Euler path or circuit. Try that!

In turns out that the converse of the theorem is true. And we will prove it by providing an algorithm for constructing an Euler circuit.

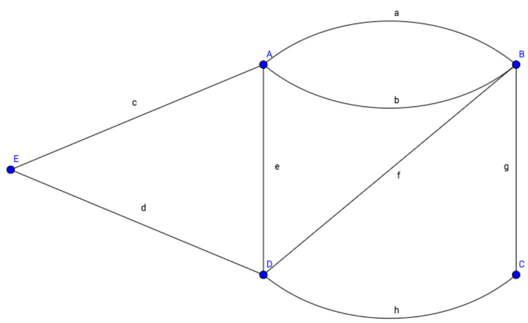
**Theorem (Euler's Theorem):** A finite connected graph in which every vertex has even degree has an Euler circuit.

What does it mean that this is the converse of the first theorem?

**Corollary:** A finite connected graph that has exactly two vertices of odd degree has an Euler path.

**Proof:** Assuming the theorem is true (which we will prove shortly) then given a graph with exactly two odd vertices, draw in an extra edge between those vertices. Now you have a graph with no odd vertices so by the theorem is has an Euler circuit. Delete that extra edge you added from the Euler circuit to get an Euler path.

## Fleury's Algorithm




Let's say we want to find an Euler circuit in this graph. We start at vertex A and travel along edge a to B. When we get to B, since there was an even number of edges and we just used one, there must be another edge to take. Maybe we choose to take b. Then Maybe we take e. At every step, once we enter a vertex, there is an odd number of edges left - so always at least one. So we just take one. So maybe we continue with h g f then d c. So we are done! Does this process always work?

Let's try again. Maybe we will start out with a b e then choose d c. Then we get suck and haven't done h and g. So when we chose d that was a bad choice. Why was it a bad choice?

Let's see what happens if you erase edges as you go. Draw the graph without edges a b or e.

Now draw it without a b e or f and again without a b e or d.

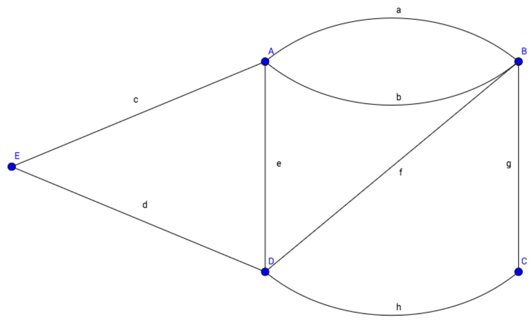
What do you notice about these two graphs that could have told you to do f next instead of d?

The ways Fleury's algorithm works is that you always have to choose an edge where is you remove it, the graph doesn't become disconnected. This is always possible but takes a lot of time. In fact, the algorithm is $\Theta(|V(G)|^2|E(G)|)$ where $|V(G)|$ is the number of vertices in the graph and $|E(G)|$ is the number of edges in the graph. Now let's look at another algorithm that is $O(|E(G)|)$.

## Another Algorithm

Here's another way to find an Euler Circuit. Start just as before where we travel around and always take an edge. We know we can always do this except with we get back to where we started. Why do we sometimes get trapped where we started?

So, we might end up with a circuit that is not an Euler Circuit. But then what's left? On the graph below, find a circuit that is not an Euler Circuit, then draw what edges are left:




What's left should still be a graph with an even number of vertices. Since the original graph we connected, you must have visited one of these vertices before. Starting at that vertex, follow the same rules. You will find another circuit. Attach that to the original circuit at the vertex you already visited. Continue this process until you have used all edges.

Here's the algorithm written out in pseudocode. It consists of two functions - ClosedPath and Euler Circuit

**ClosedPath(graph,vertex)**

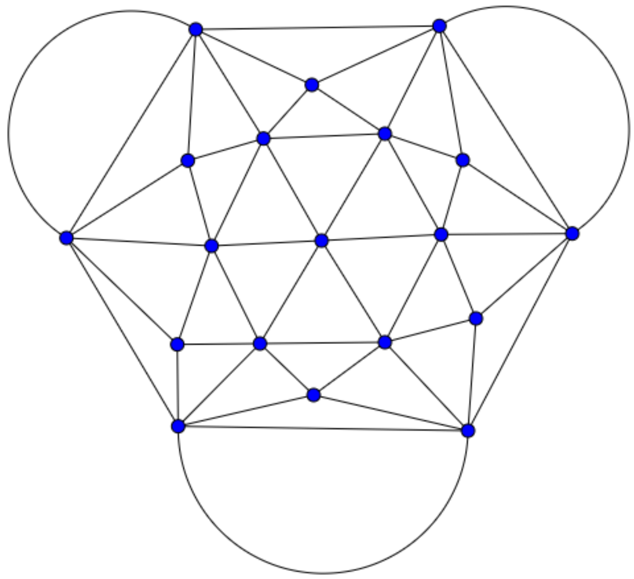
{Input: A graph H in which every vertex has even degree and a vertex v of positive degree.}
{Output: A simple closed path P though vertex v}
Choose an edge $e$of H with endpoint v.
Let P:=$e$. Remove $e$from E(H).
`while` there is an edge at terminal vertex of P `do`
Choose such an edge, add it to the end of P and remove it from E(H)
`return` P

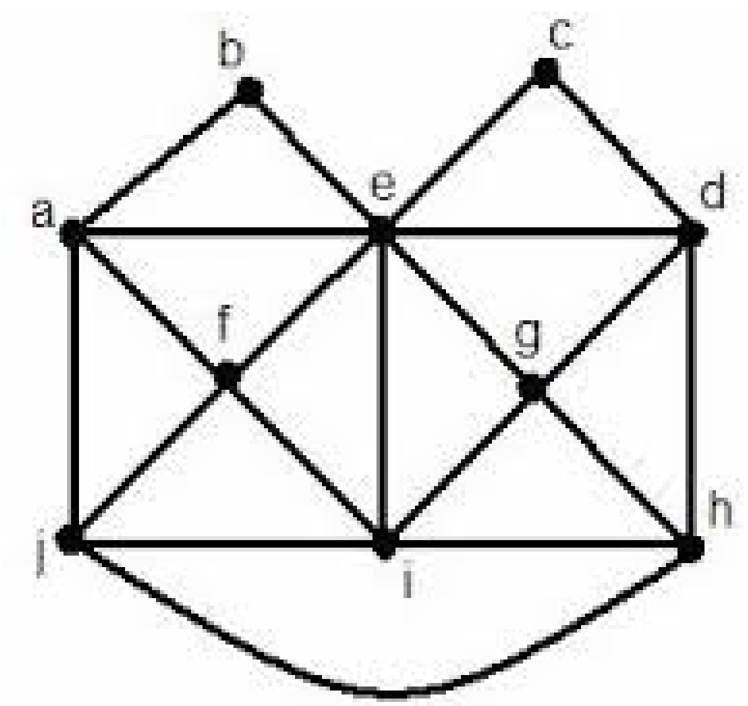
**EulerCircuit(graph)**

{Input: A connected graph G with all vertices of even degree}
{Output: An Euler Circuit C of G}
Choose $v \in V(G)$
Let C:=ClosedPath(G, v)
`while` length(C)<|E(G)| `do`
Choose a vertex w on C of positive degree in $G\backslash C$.
Attach ClosedPath($G\backslash C$,w) to C to w to obtain a longer simple closed path C.
`return` C

Discuss what is going on with your group!

Use the algorithm to find an Euler Circuit on the graphs below.






Finally: Play some One Touch Draw or Planarity!!