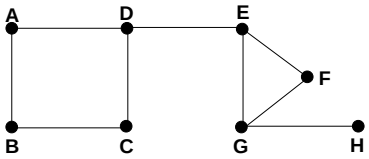
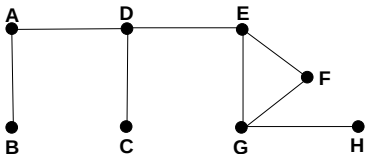
Sec 2. Euler Circuits, cont.

- Euler Circuits traverse each edge of a connected graph exactly once.
- Recall that all vertices must have even degree in order for an Euler Circuit to exist.
- **Fleury's Algorithm** is a method for finding an Euler Circuit.
- A **cut edge** in a graph is an edge whose removal disconnects a component of the graph.

Which are cut edges?



Graph I.

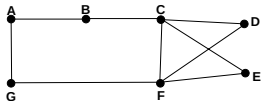


Graph II.

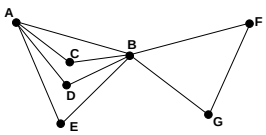
Fleury's Algorithm

1. Pick a vertex and write it down as the start of the Euler Circuit.
2. Pick any edge incident to this vertex, traverse it to the other end point, add that end point to the Circuit, then erase the edge.
3. Choose any edge from this new vertex, subject to the condition that you **do not choose a cut edge** unless there is no other option.
4. Traverse this edge to the other vertex, add that vertex to the Circuit, then erase the edge.
5. Repeat from Step 3 until all the edges have been erased and you're back to the Start vertex.

Apply Fleury's Algorithm to the following graphs:



Graph I.



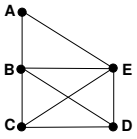
Graph II.

If we start with a connected graph in which all vertices have even degree, then Fleury's Algorithm will always produce an Euler circuit.

Section 3. Hamilton Circuits

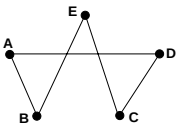
A **Hamilton Circuit** in a graph is a circuit that visits each **vertex** exactly **once**, returning to the starting vertex to complete the circuit.

Which of the given paths are Hamilton Circuits?



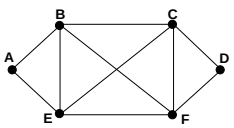
a) A – E – C – D – E – B – A
b) A – E – C – D – B – A
c) D – B – E – A – B
d) E – D – C – B – E

Graph I.



a) A – B – C – D – E – A
b) B – E – C – D – A – B
c) E – B – A – D – A – D – C

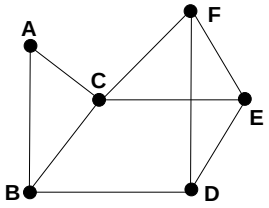
Graph II.



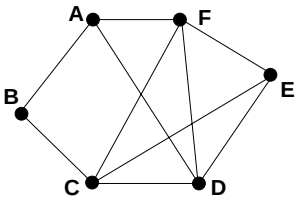
Graph III.

A – B – E – D – C – F – A
A – B – C – D – E – F – C – E – B – F – A
B – C – D – E – F – B

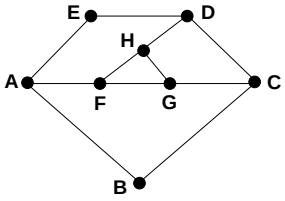
Find a Hamilton Circuit in the following graphs (if one exists):



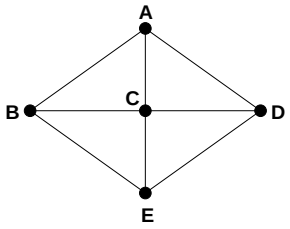
Graph I.



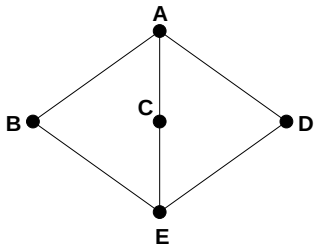
Graph II.



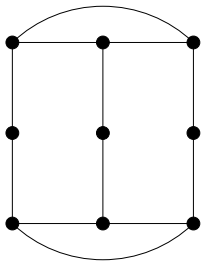
Graph III.



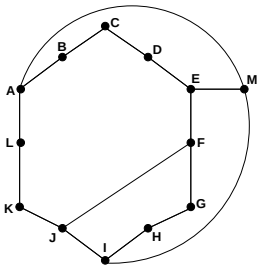
Graph IV.



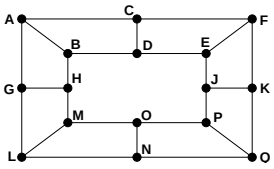
Graph V.



Graph VI.



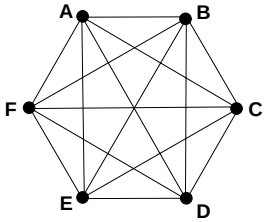
Graph VII.



Graph VIII.

Hamilton Circuits for Complete Graphs

Any complete graph with three or more vertices has a Hamilton Circuit.



When Hamilton Circuits Are the Same: Hamilton circuits that differ ***only*** in their starting points will be considered to be the same circuit.

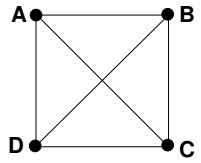
A–B–C–D–E–F–A is essentially the same as **B–C–D–E–F–A–B**, since the vertices are visited in the same order.

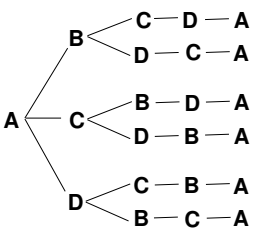
Tree Diagrams

Tree Diagrams are a convenient strategy for inspecting all possible combinations when we are given multiple choices to make.

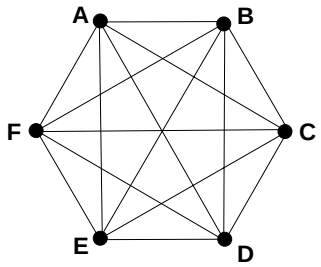
In **Hamilton Circuits**, we can begin with any vertex in the graph, then choose among those which are adjacent to it...

In the graph below, note how the tree diagram lists all the possible circuits in the graph: 3! (three factorial or $1 \times 2 \times 3$) of them.





How many circuits would we find in the following graph?



How many circuits would a complete graph over 10 vertices, a K_{10}, have?

Number of Hamilton Circuits in a Complete Graph

A complete graph with n vertices has $(n-1)!$ Hamilton circuits.

Recall: a **weighted graph** is one in which weights or costs are assigned to the graph's edges.

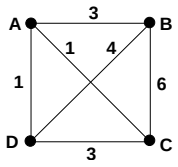
The **total weight** of a circuit in a weighted graph is the sum of the weights of the edges in the circuit.

A **Minimum Hamilton Circuit** in a weighted graph is a Hamilton Circuit with **smallest possible total weight**.

Brute Force Minimum Hamilton Circuit Algorithm

1. Choose a starting vertex
2. List all the Hamilton Circuits with that starting vertex
3. Find the total weight of each circuit
4. Choose a Hamilton Circuit with the smallest total weight

Find a minimum Hamilton Circuit for the complete, weighted graph shown here:



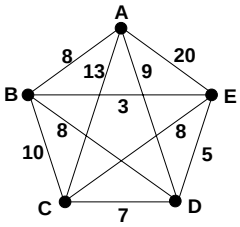
Possible Circuits	Total circuit weight

- This brute force method is **time consuming**.
- As the number of vertices grows, the time it takes to check *all* possible Hamilton Circuits grows **very** fast!
- Thus, we're interested in other methods for finding a solution — in this case, just a reasonably good solution most of the time.
- This type of algorithm is called an **approximation algorithm**.
- The following algorithm finds an **approximate** solution to the Minimum Hamilton Circuit problem.

Nearest Neighbor Algorithm

1. Choose a starting vertex for the circuit; call it vertex **A**.
2. Check all edges incident to **A**, and choose the one that has smallest weight; proceed along this edge to the next vertex.
3. At each vertex you reach, check the edges from there to **vertices not yet visited**. Choose one with smallest weight. Proceed along this edge to the next vertex.
4. Repeat Step 3 until all vertices have been visited.
5. Return to the starting vertex **A**.

Apply this algorithm to the following graph. Does it make any difference at what vertex we begin?



Starting Vertex	Circuit Using Nearest Neighbor	Total Weight
A		
B		
C		
D		
E		

Find a Hamilton Circuit starting at vertex **A** in the following graph: