Fleury's Algorithm

Let G be a connected graph. If G is Eulerian then Fleury's algorithm will produce an Eulerian trail in G.

In a connected graph G, a **bridge** is an edge which, if removed, produces a disconnected graph.

Fleury's Algorithm

Let G be an Eulerian graph.

STEP 1: Choose any vertex v of G and set *current vertex* equal to v and *current trail* equal to the empty sequence of edges.

STEP 2: Select any edge e incident with the *current vertex* but choosing a bridge only if there is no alternative.

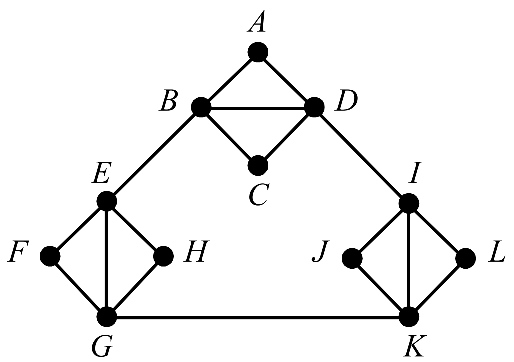
STEP 3: Add e to the *current trail* and set the *current vertex* equal to the vertex at the 'other end' of e. [If e is a loop, the *current vertex* will not move.]

STEP 4: Delete e from the graph. Delete any isolated vertices.

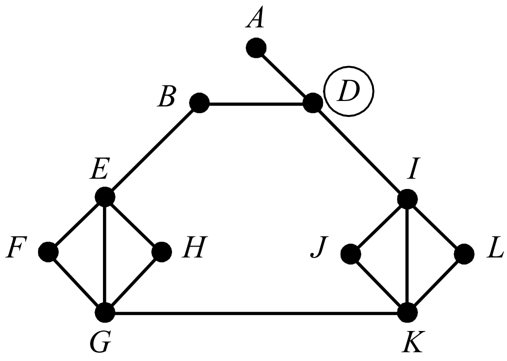
Repeat steps 2 – 4 until all edges have been deleted from G. The final *current trail* is an Eulerian trail in G.

Example

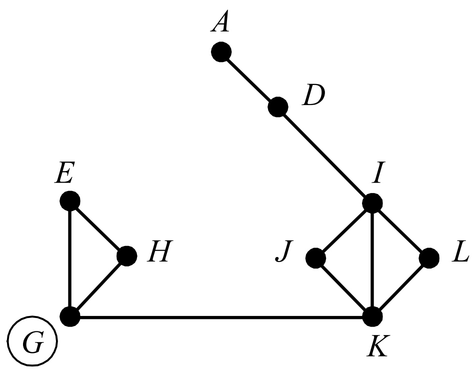
Apply Fleury's algorithm, beginning with vertex A, to find an Eulerian trail in the following graph. In applying the algorithm, at each stage chose the edge (from those available) which visits the vertex which comes first in alphabetical order.



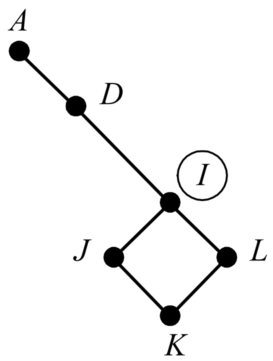
Starting at A, choose AB, BC, CD. This gives the following graph (with the current vertex circled).



The edge DA is a bridge; choose DB, BE, EF, FG to produce the following graph.



(GK is a bridge.) Choose GE, EH, HG, GK, KI to give the following.



The edge ID is a bridge so choose IJ followed by JK, KL, LI, ID, DA.

The complete trail is:

$$A\ B\ C\ D\ B\ E\ F\ G\ E\ H\ G\ K\ I\ J\ K\ L\ I\ D\ A.$$