

Graph Theory

Glossary of Terms

Adjacent Vertices: Two vertices are adjacent if they are joined by a single edge.

Adjacent Edges: Two edges are adjacent if they are joined by a vertex.

Circuits: A circuit is a path which starts and ends at the same place.

Connected Graph: A graph is connected if you can get from any vertex to any other vertex by travelling over the edges of the graph. (Otherwise we call the graph disconnected.)

Degree of a Vertex: The degree of a vertex is the number of edges that meet at that vertex.

Euler Circuit: An Euler Circuit is an Euler Path which begins and ends at the same vertex.

Euler Path: An Euler Path is a path in which every edge is traveled exactly once.

Even Vertex: An even vertex is a vertex whose degree is an even number.

Odd Vertex: An odd vertex is a vertex whose degree is an odd number.

Path: A path is a sequence of vertices with the property that each vertex is adjacent to the next one. Note that each edge may only appear in a path once. (The length of a path is the number of edges travelled on that path.)

Theorems

Euler's Theorem 1

- If a graph has *any* odd vertices, then it *cannot* have an Euler circuit.
- If a graph is *connected* and *every* vertex is an even vertex, then it has at least one Euler circuit (and usually more).

Euler's Theorem 2

- If a graph has *more than two* odd vertices, then it cannot have an Euler path.
- If a graph is connected and has exactly *two* odd vertices, then it has at least one Euler path (and usually more). Any such path must start at one of the odd vertices and end at the other one.

Euler's Theorem 3

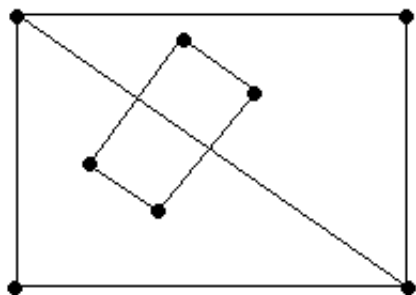
- The sum of the degrees of all the vertices of a graph equals twice the number of edges (and therefore is an even number).
- A graph always has an even number of *odd* vertices.

Fleury's Algorithm for Finding an Euler Circuit:

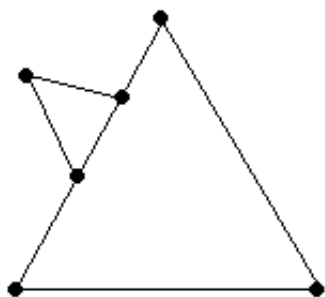
- First makes sure that the graph is connected and all the vertices have even degree.
- Pick any vertex as the starting point.
- When you have a choice, always choose to travel along an edge that is not a bridge of the yet-to-be-traveled part of the graph.
- Label the edges in the order in which you travel them.
- When you can't travel any more, *stop*. You are done!

Examples

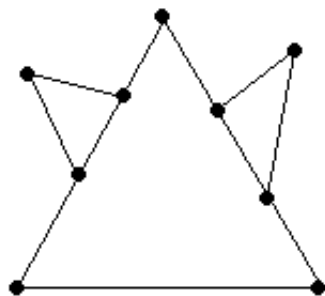
Example 1:



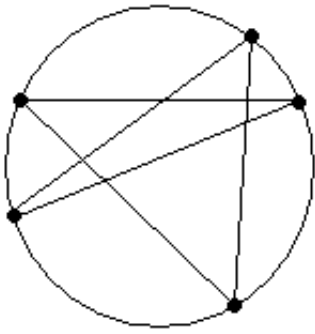
Example 2:



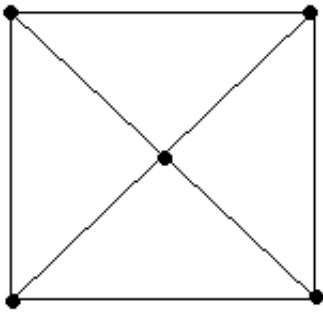
Example 3:



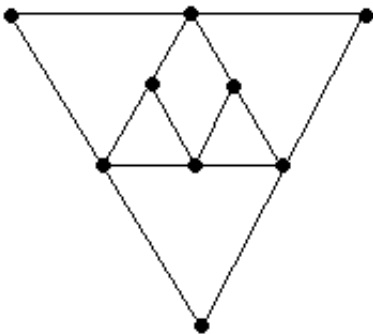
Example 4:



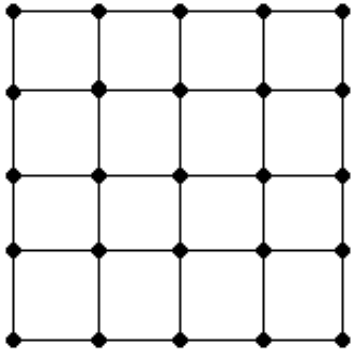
Example 5:



Example 6:



Example 7:



Notes: