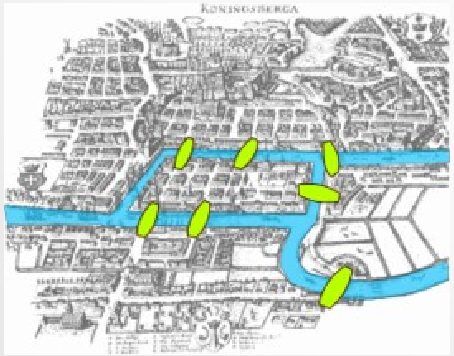
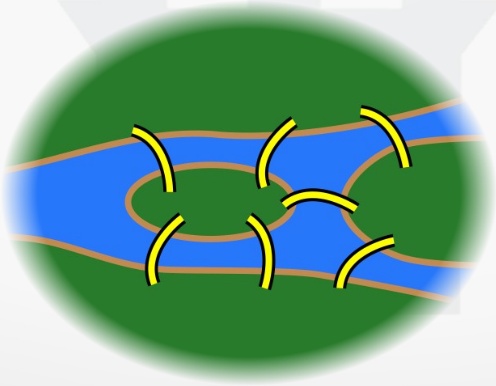
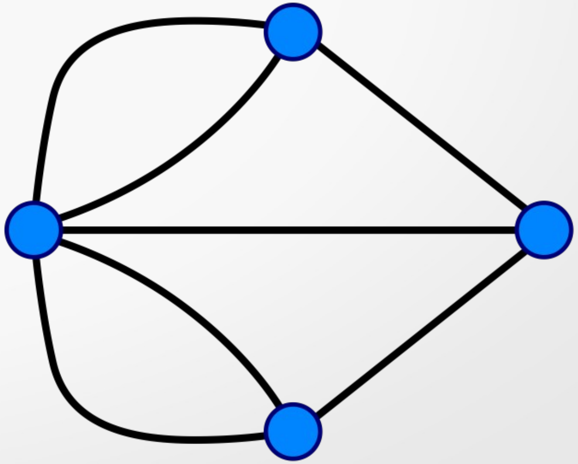
Paths & Circuits

The Seven Bridges of Königsberg

- Can you take a walk and visit all bridges exactly once?
  - You don’t have to end up where you began
- Leonhard Euler 1735 (interestingly, Benjamin Franklin was a contemporary, almost same birth/death)
  - Map physical environment onto a graph; the beginning of graph theory
  - Interesting lecture about Euler: https://www.youtube.com/watch?v=h-DV26x6n_Q



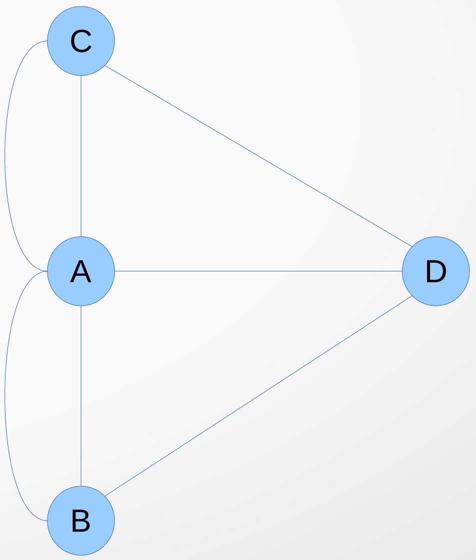




Euler Paths and Circuits

- An ***Euler path*** is a path using every edge of the graph G exactly once
- An ***Euler circuit*** is an Euler path that returns to its start

Does this graph have an Euler circuit?

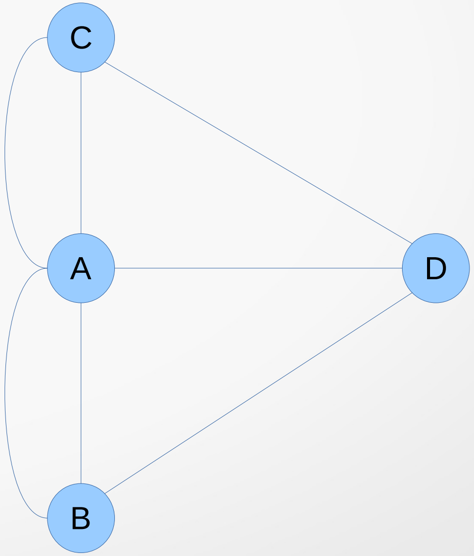


Euler Paths and Circuits

- An ***Euler path*** is a path using every edge of the graph G exactly once
- An ***Euler circuit*** is an Euler path that returns to its start

Does this graph have an Euler circuit? No

but we used trial and error to figure this out

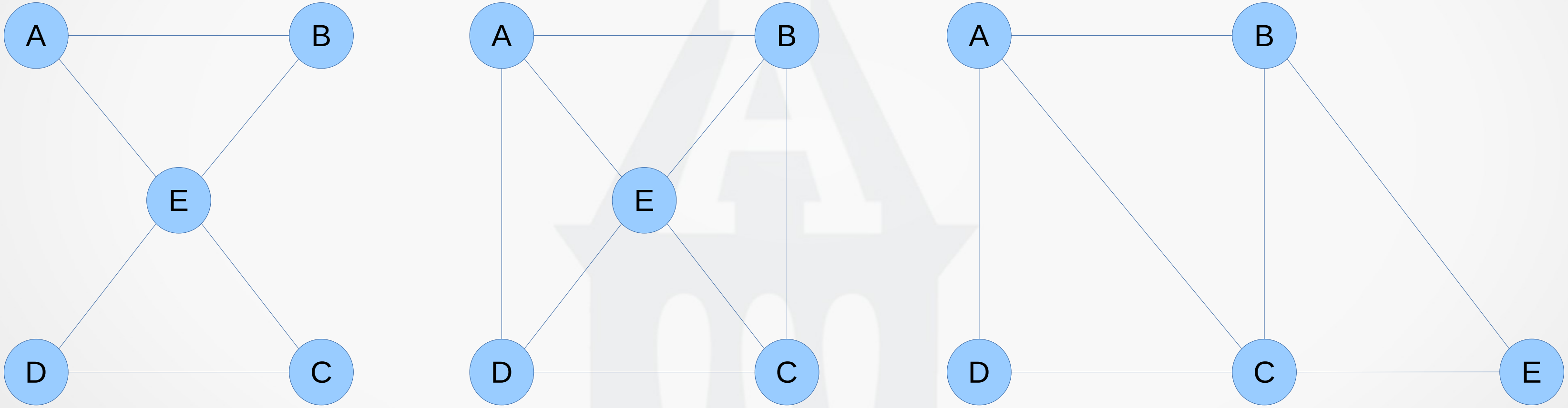


Euler Paths and Circuits

- May be a multigraph
  - More than one connection between same two vertices
- A ***connected multigraph*** has an ***Euler circuit*** *iff each of its vertices has an even degree*
- A ***connected multigraph*** has an ***Euler path*** but ***not an Eurler circuit*** *iff it has exactly two vertices of odd degree*

Euler Paths and Circuits – Undirected Graphs

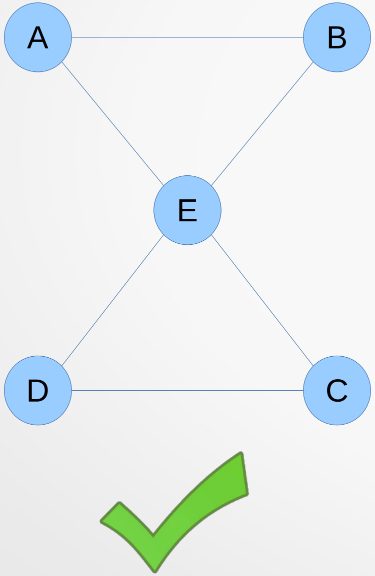
- Which of these graphs has an Euler ***path***?

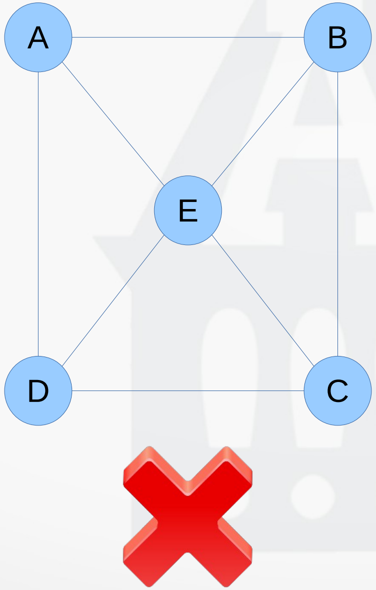


Euler Paths and Circuits – Undirected Graphs

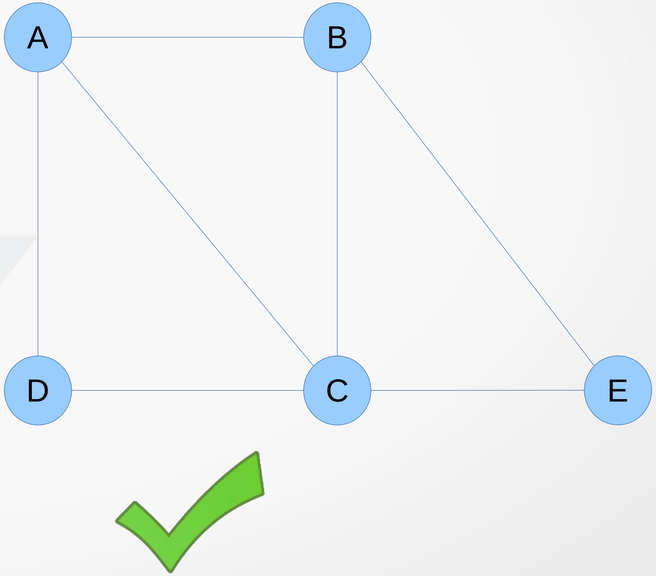
- Which of these graphs has an Euler ***path***?

a, e, c, d, e, b, a



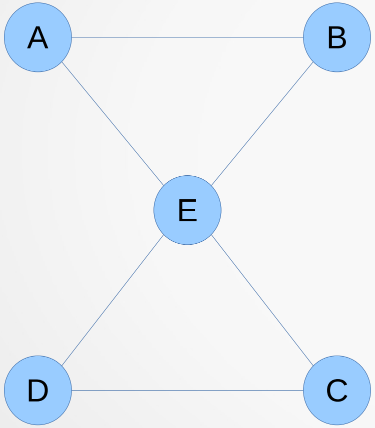


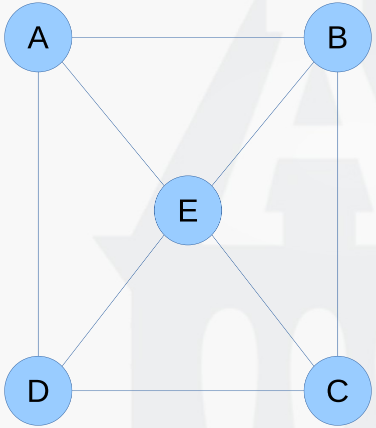
a, d, c, e, b, a, c

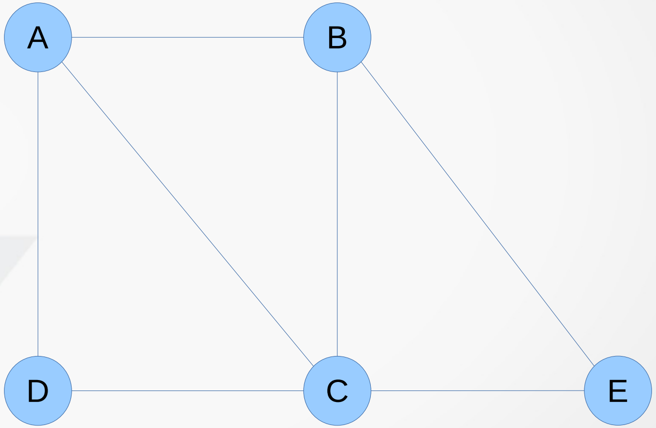


Euler Paths and Circuits – Undirected Graphs

- Which of these graphs has an Euler ***circuit***?



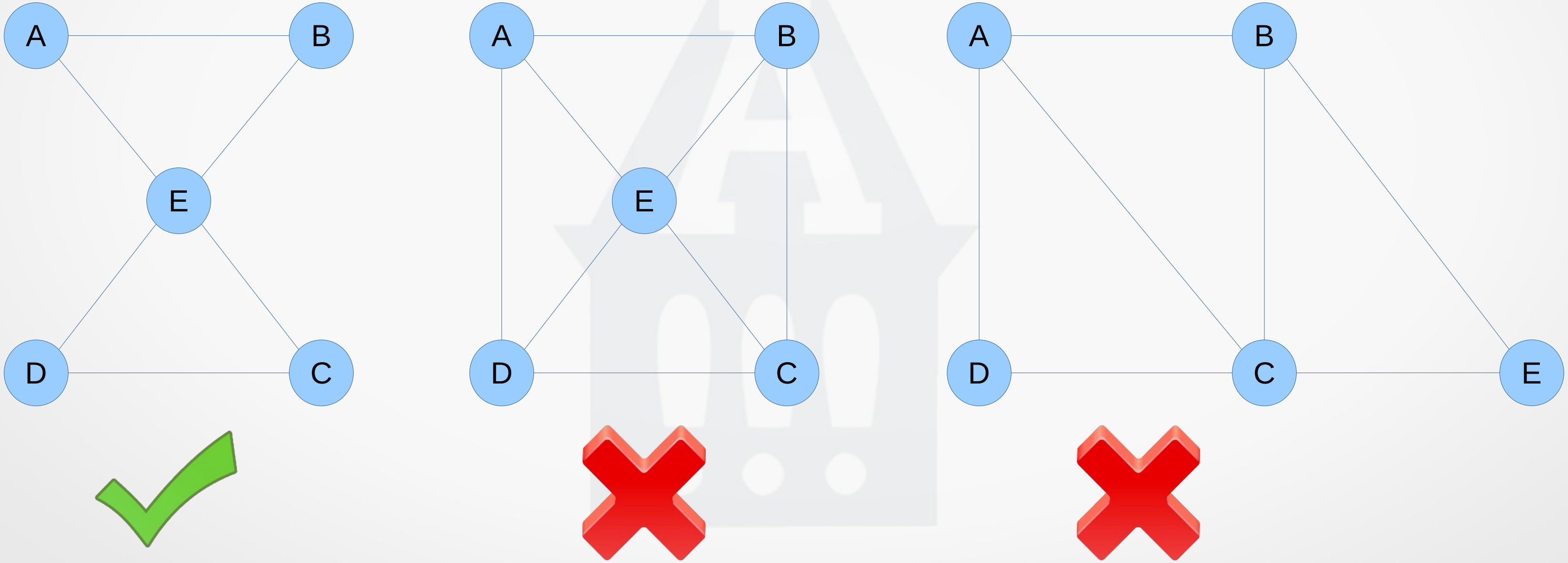




Euler Paths and Circuits – Undirected Graphs

- Which of these graphs has an Euler ***circuit***?

a, e, c, d, e, b, a



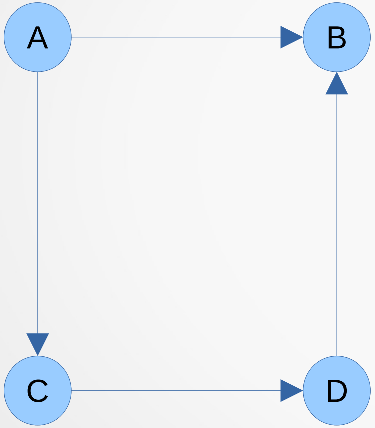
Euler Paths and Circuits – Directed Graphs

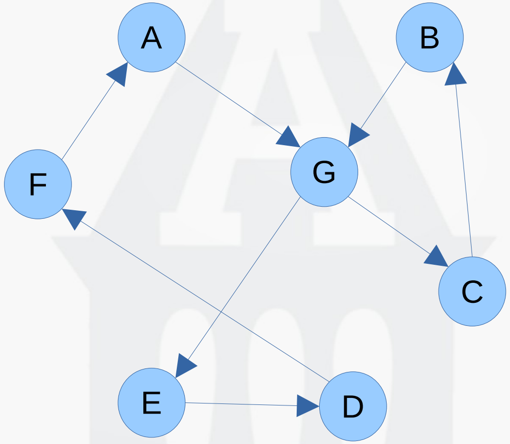
- Can do the same thing with directed graphs
- Have to count the in-degree and out-degree at each vertex

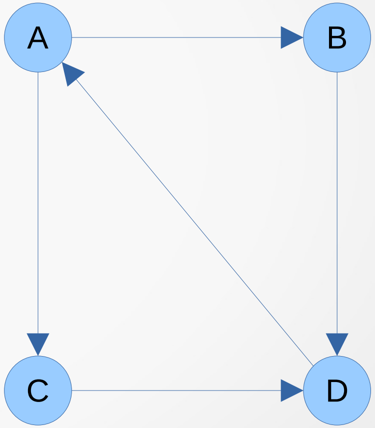
- There is an Euler path iff
  - at most one vertex has (out-degree – in-degree) = 1
  - at most one vertex has (in-degree – out-degree) = 1
  - every other vertex has in-degree equal to the out-degree
- There is an Euler circuit iff
  - the in-degree is equal to the out-degree for every vertex

Euler Paths and Circuits – Directed Graphs

- Which of these graphs has an Euler ***path***?

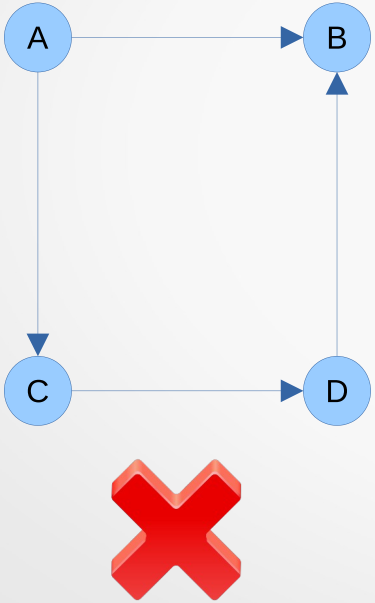




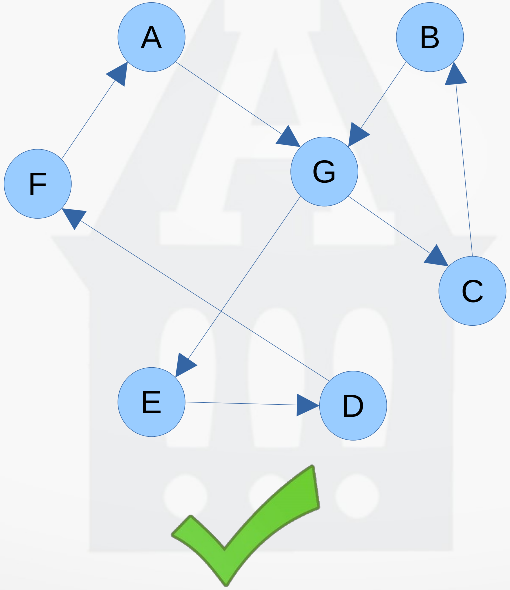


Euler Paths and Circuits – Directed Graphs

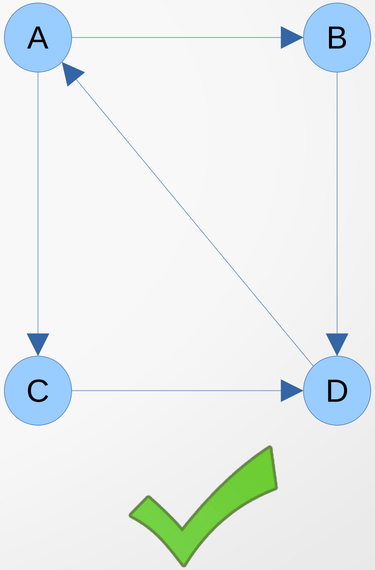
- Which of these graphs has an Euler ***path***?



a, g, c, b, g, e, d, f

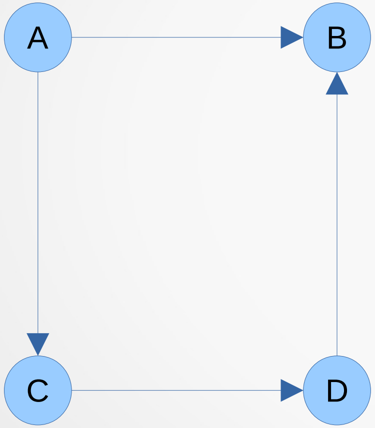


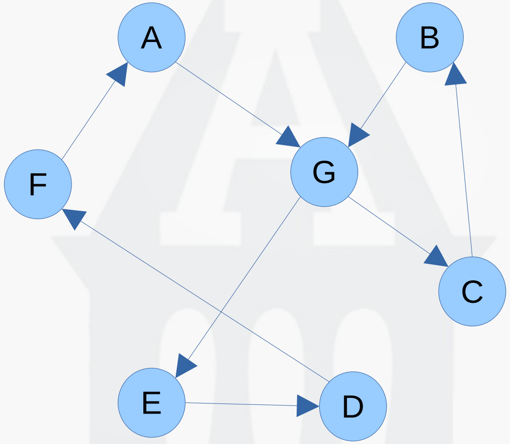
a, c, d, a, b, d

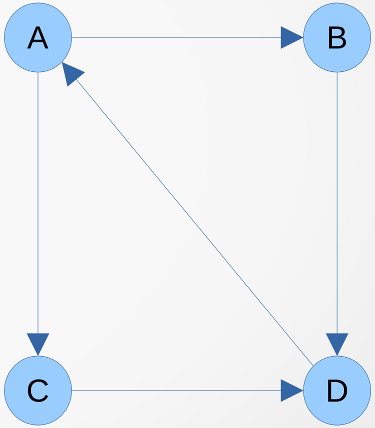


Euler Paths and Circuits – Directed Graphs

- Which of these graphs has an Euler ***circuit***?



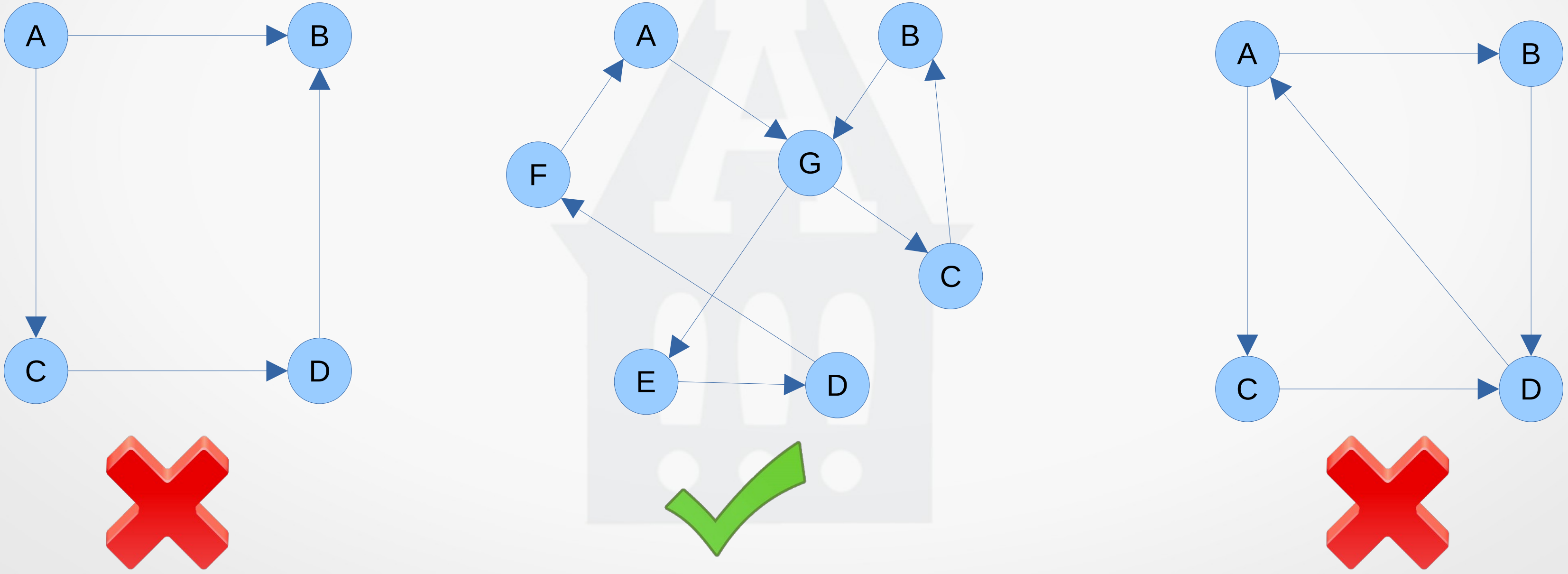




Euler Paths and Circuits – Directed Graphs

- Which of these graphs has an Euler ***circuit***?

a, g, c, b, g, e, d, f, a



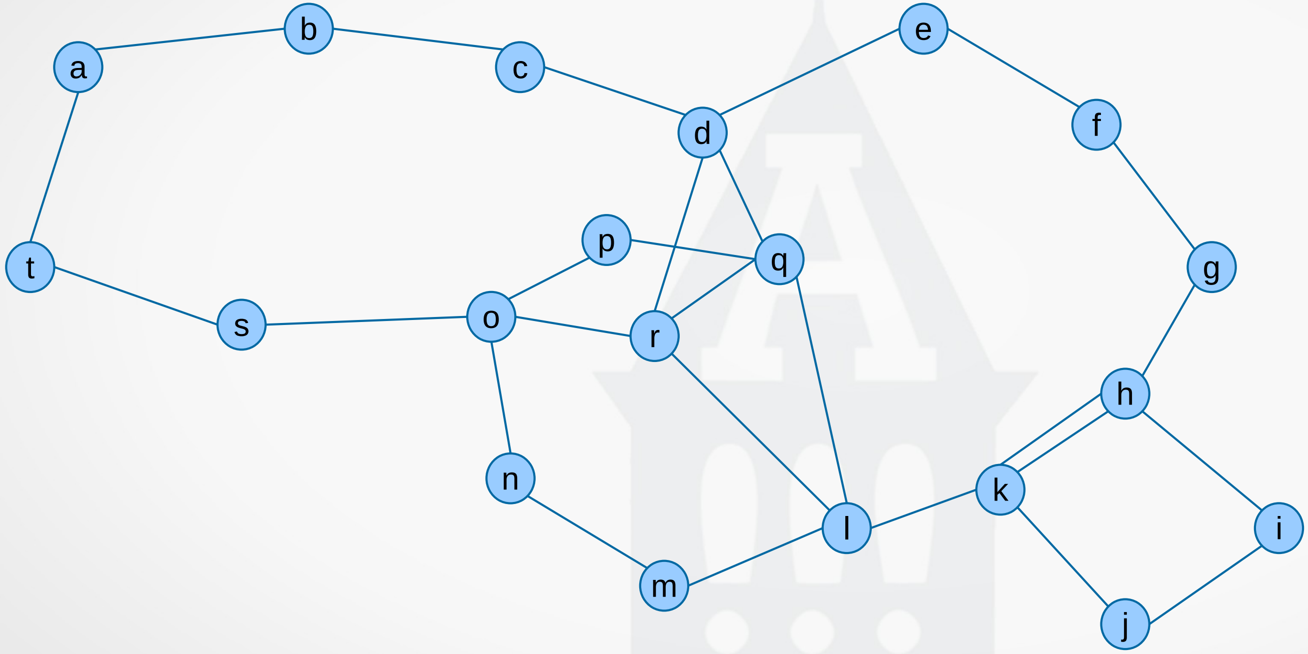
Euler Paths and Circuits – Applications

- Determining routes for things like…
  - Snow removal
  - Inspecting railroad tracks
  - Airline routes
  - Anything that needs to traverse a route and be efficient in doing so
- DNA Sequencing & fragment assembly

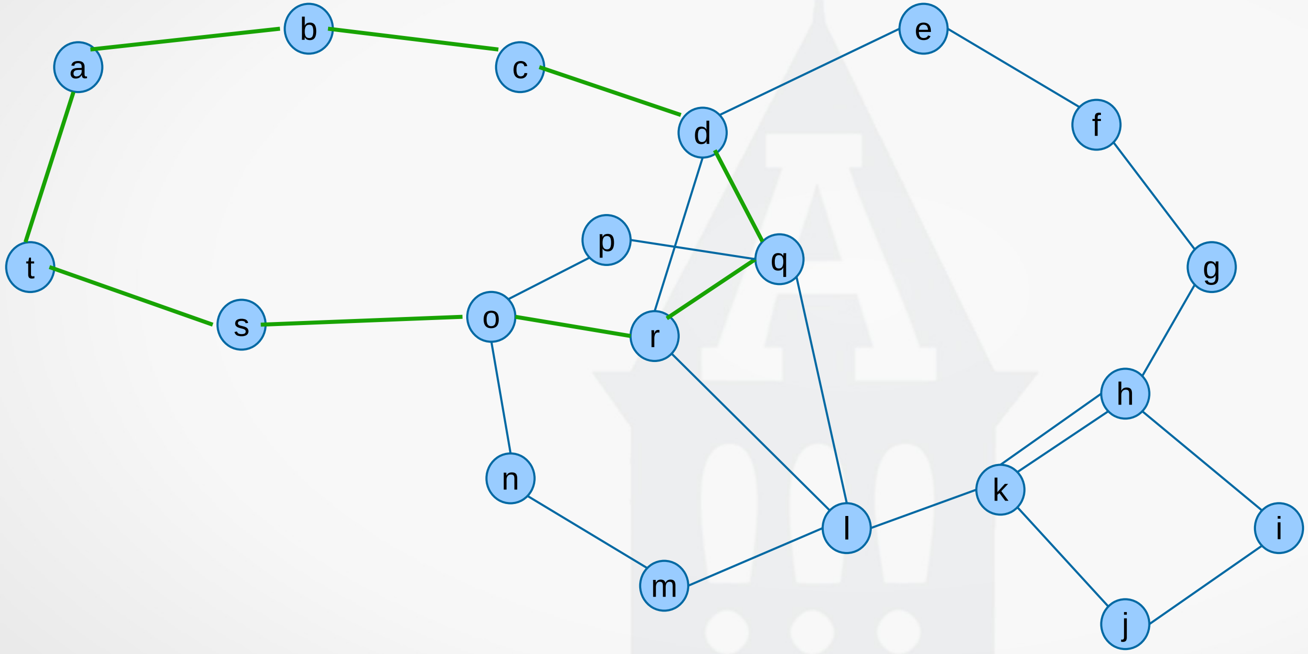
Euler Circuit – Algorithm

- Phase 1: Mark Cycles
  - Start at any node that has edges not part of a cycle already (unlabeled edges)
  - Randomly start following unlabeled edges; mark each edge used as being in the same cycle
  - Eventually you will return to the starting node
  - If all edges have been visited, you are done
    - Otherwise, repeat step 1, finding a new cycle; repeat until every edge belongs to a cycle

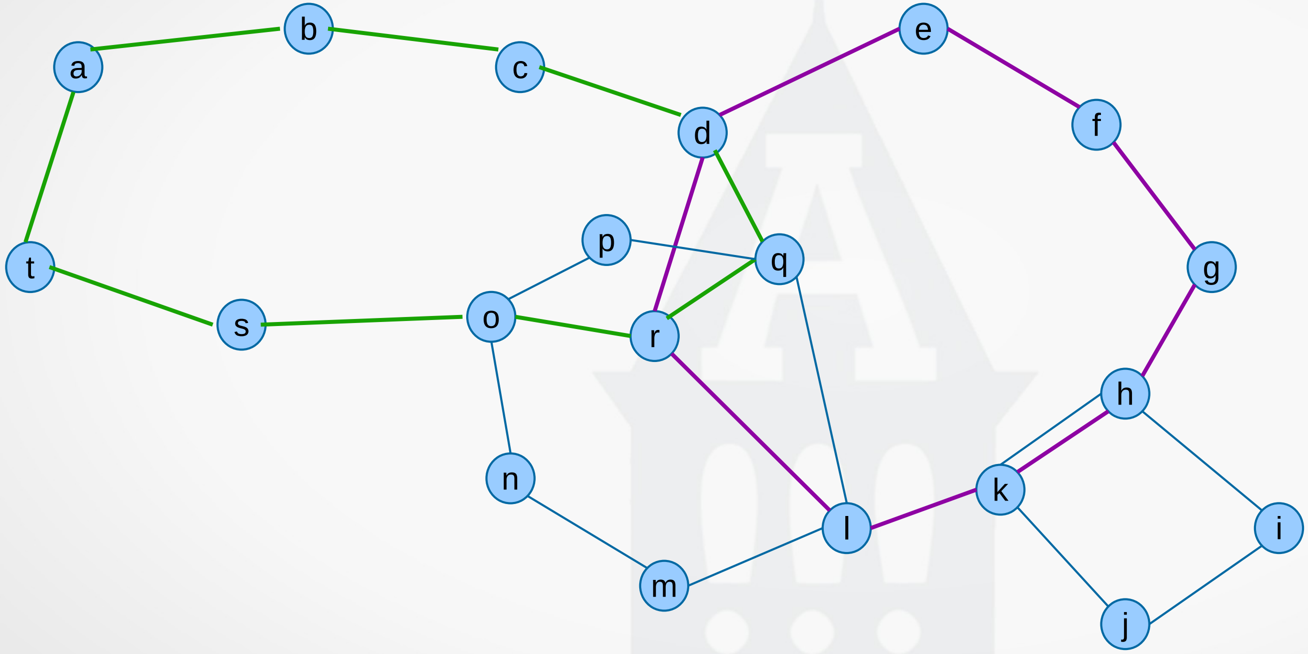
Euler Circuit – Algorithm : Marking Cycles



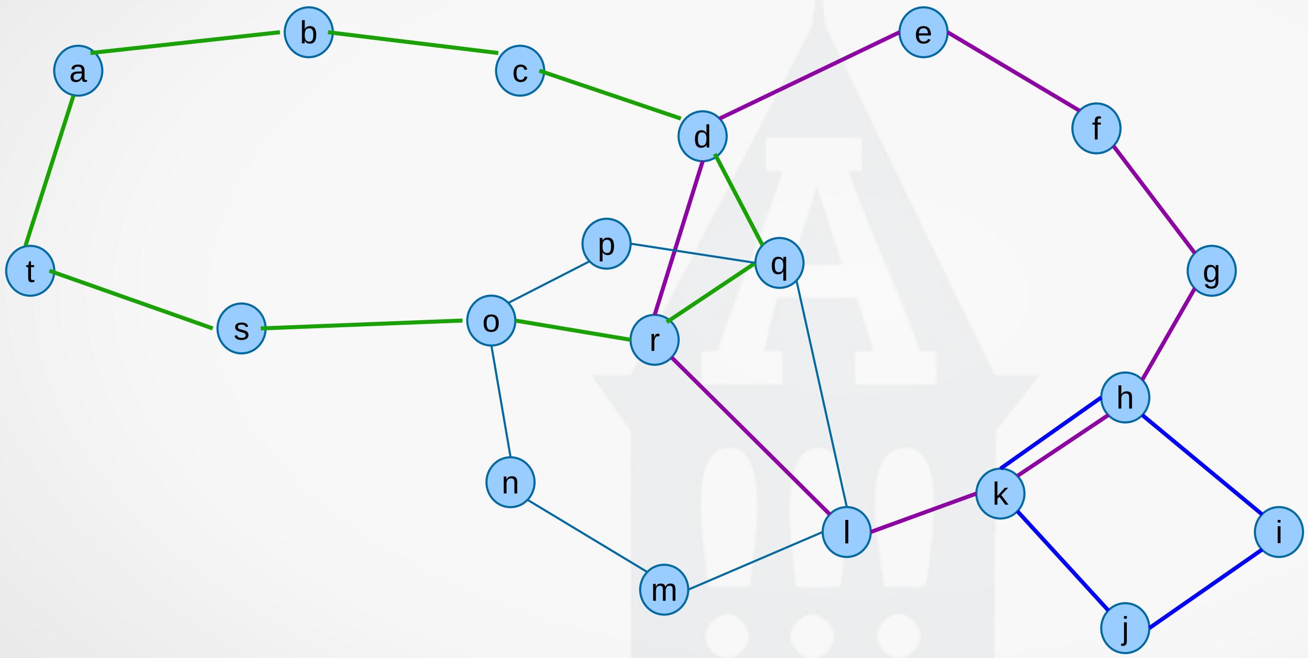
Euler Circuit – Algorithm : Marking Cycles



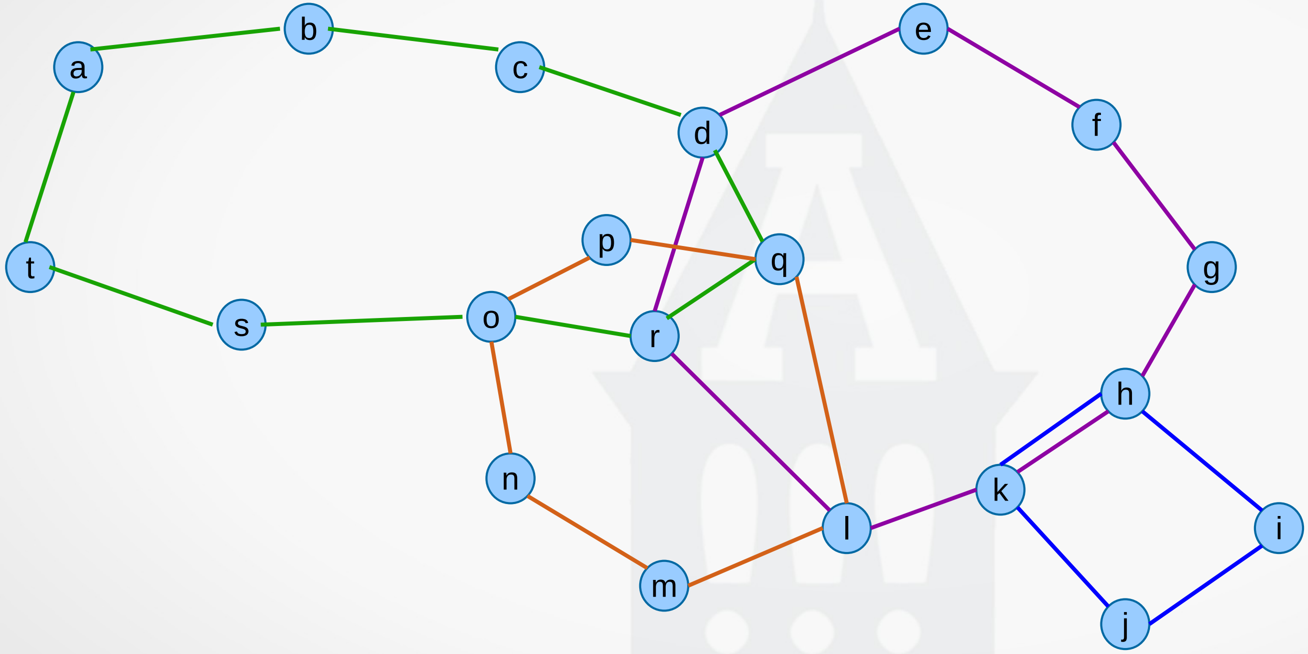
Euler Circuit – Algorithm : Marking Cycles



Euler Circuit – Algorithm : Marking Cycles



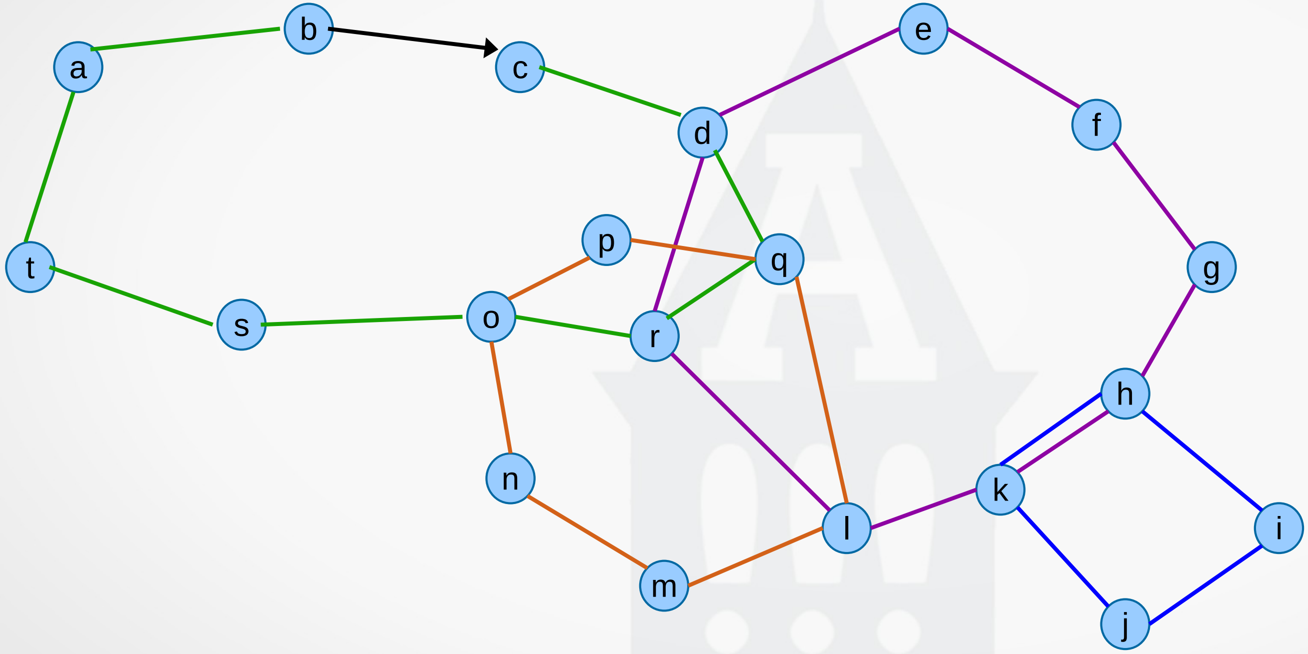
Euler Circuit – Algorithm : Marking Cycles



Euler Circuit – Algorithm

- Phase 2: Joining Cycles
  - Start traversing the cycles and join them
    - keep a stack of cycles being followed; the top of the stack indicates which cycle is currently being followed
  - As an edge is used, mark it as used (in the final circuit)
  - Beginning with a node in cycle x (put x on the stack)
  - Keep following edges of cycle x, until a node with edges of a different cycle is encountered
    - When an edge is followed, it is added to the circuit
  - Put the new cycle name on the stack and start following (and adding to the circuit) edges of the new cycle
  - If an “old cycle” is encountered, do not follow it until the current cycle is completed
    - Then pop the current cycle off the stack and start following edges of the top stack cycle
  - Eventually you will return to the starting cycle

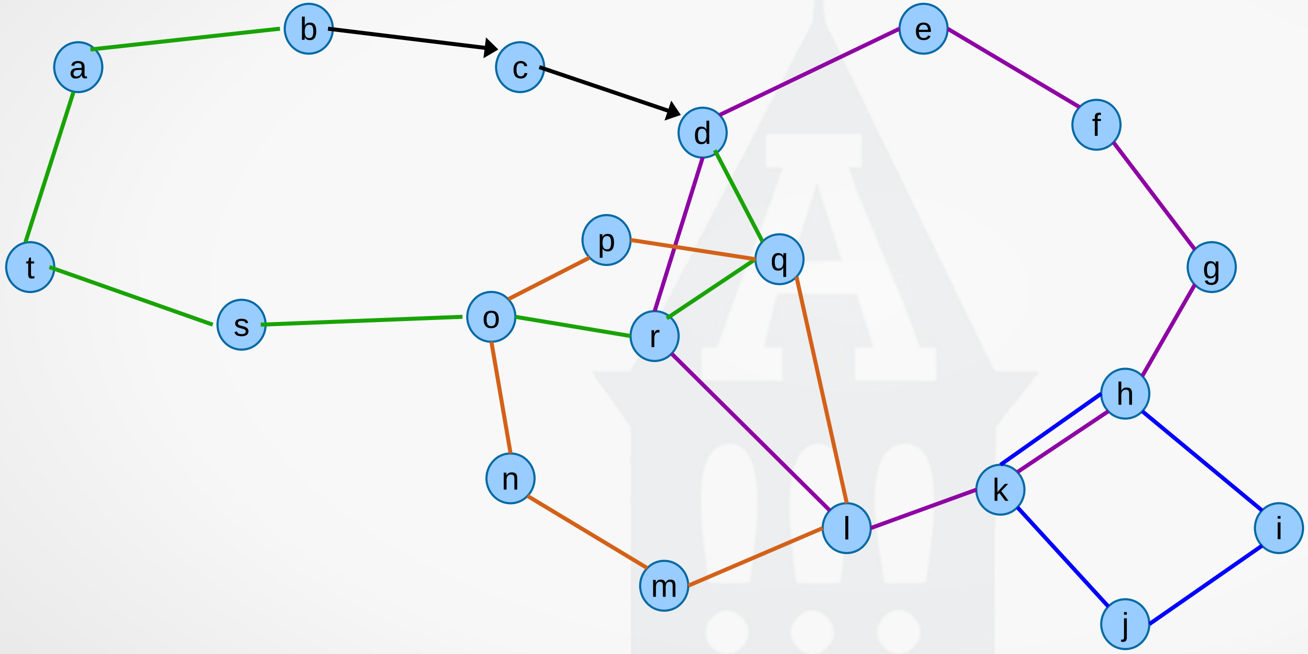
Euler Circuit – Algorithm : Joining Cycles



green

start with any edge

Euler Circuit – Algorithm : Joining Cycles

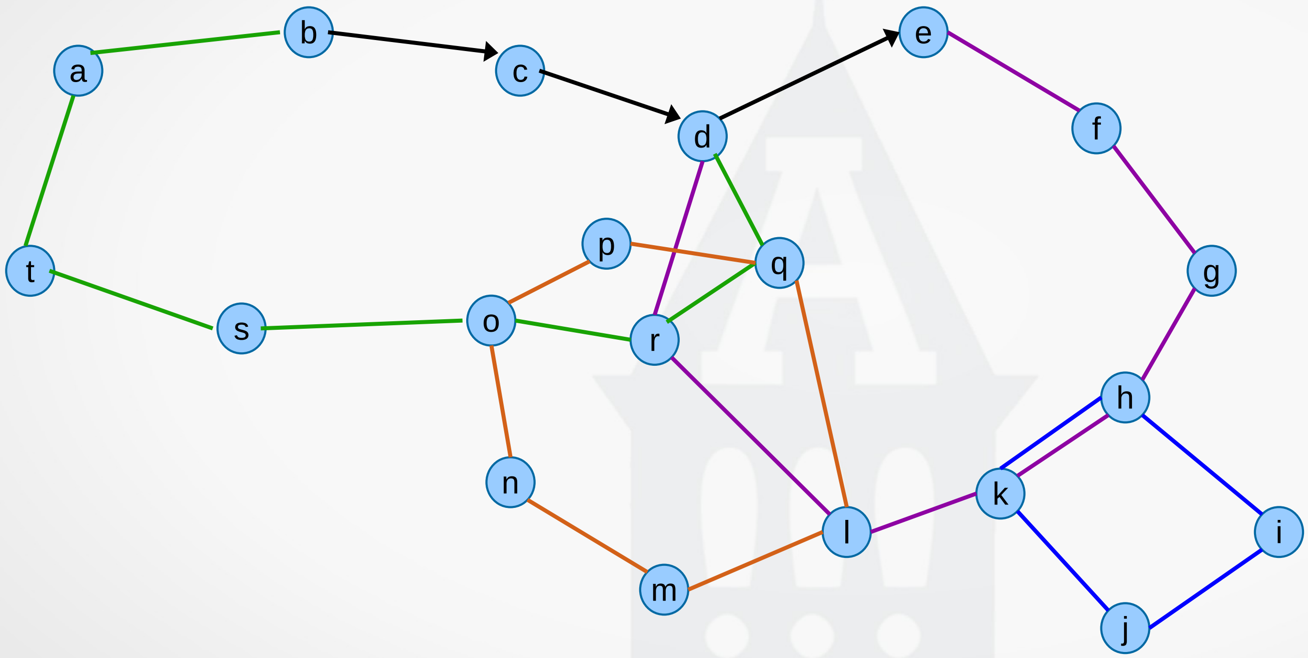


purpose

green

continue until new cycle is encountered

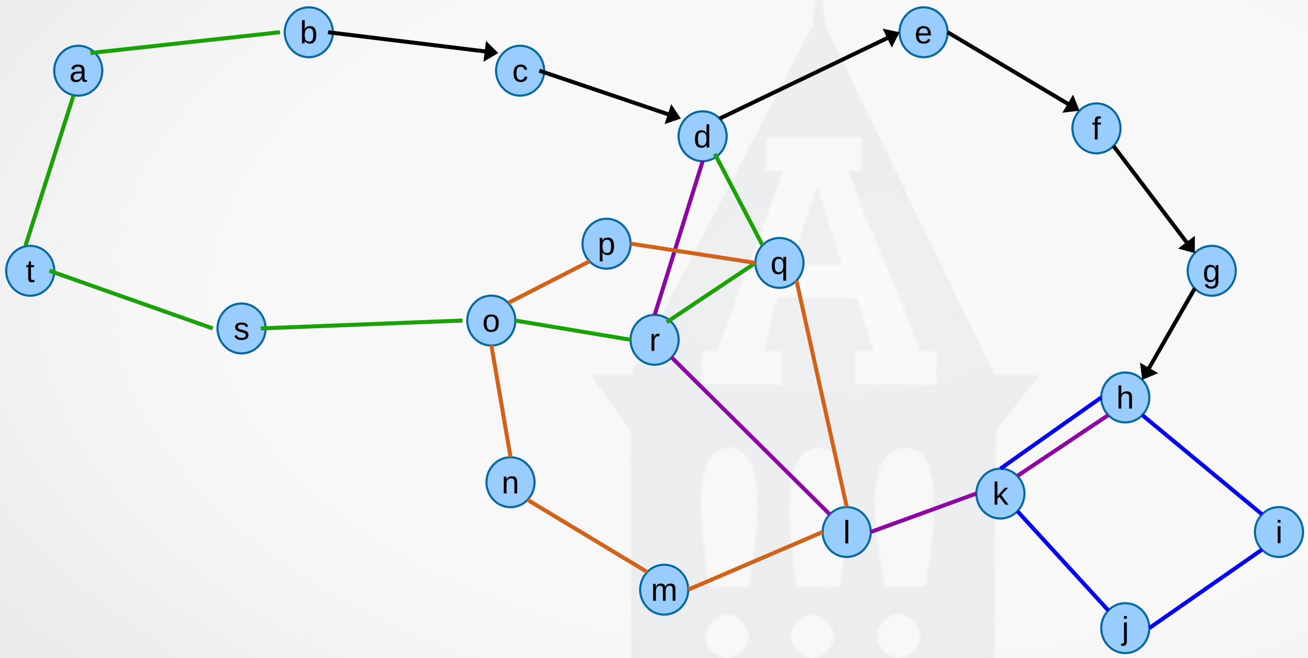
Euler Circuit – Algorithm : Joining Cycles



purple

green

Euler Circuit – Algorithm : Joining Cycles

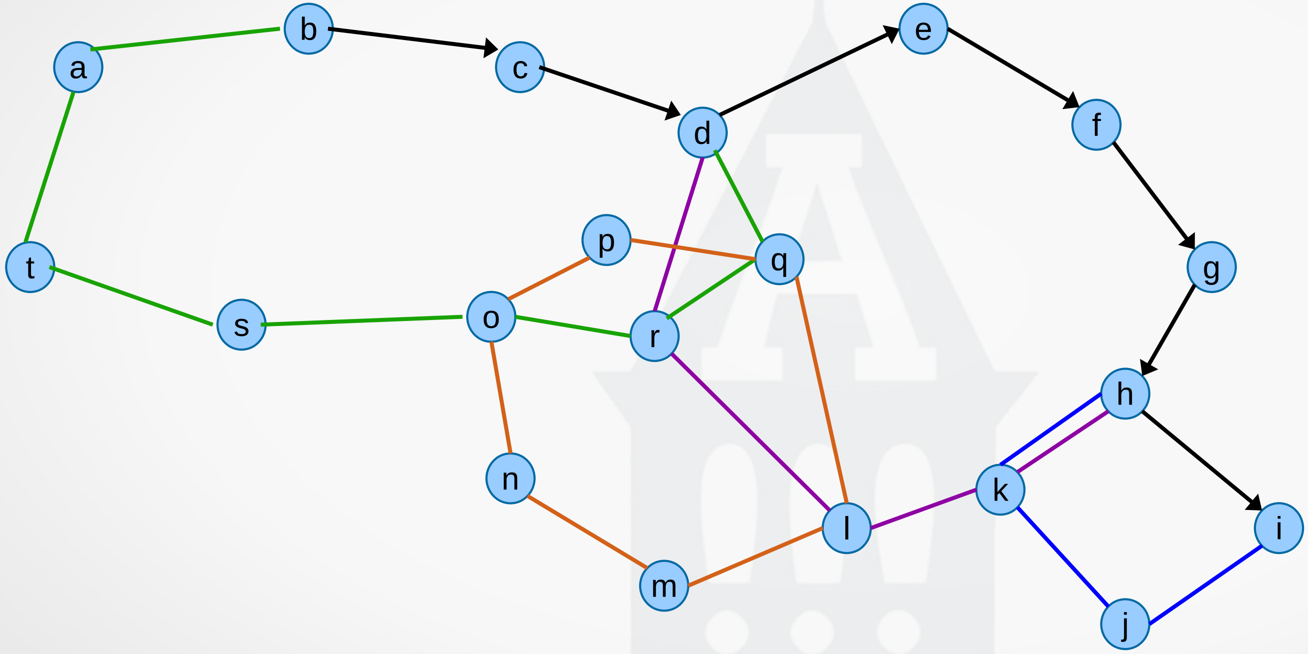


blue cycle is encountered

purple

green

Euler Circuit – Algorithm : Joining Cycles

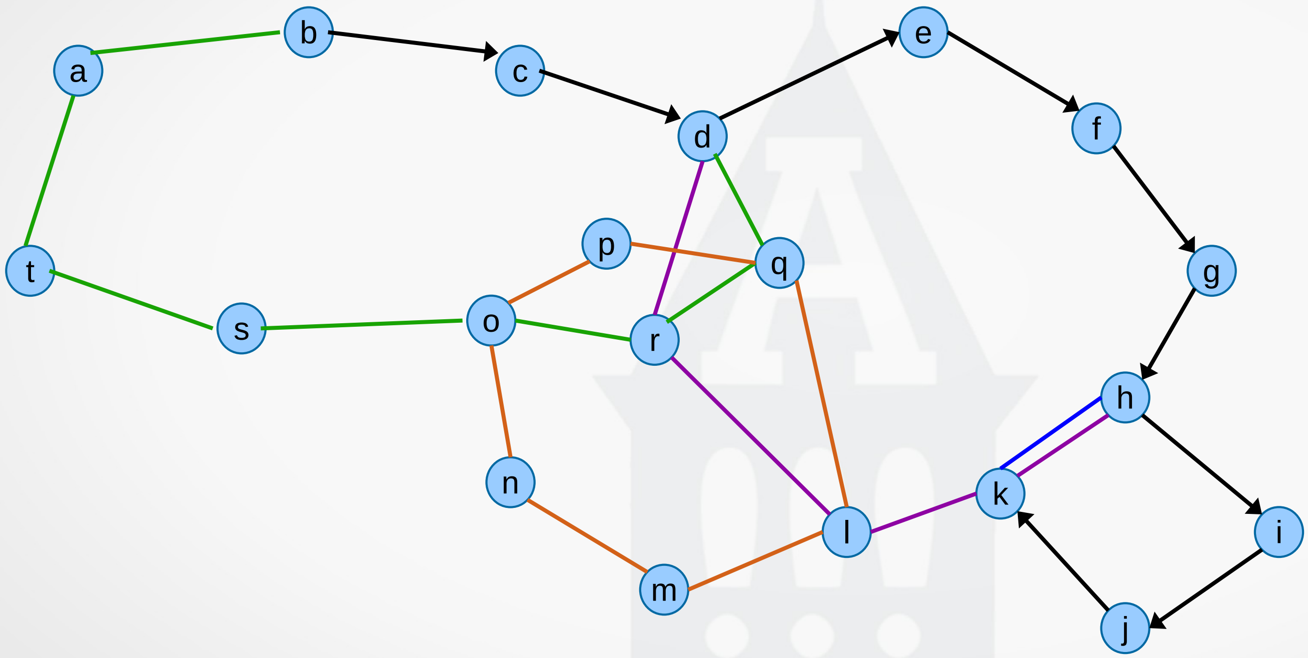


blue

purple

green

Euler Circuit – Algorithm : Joining Cycles



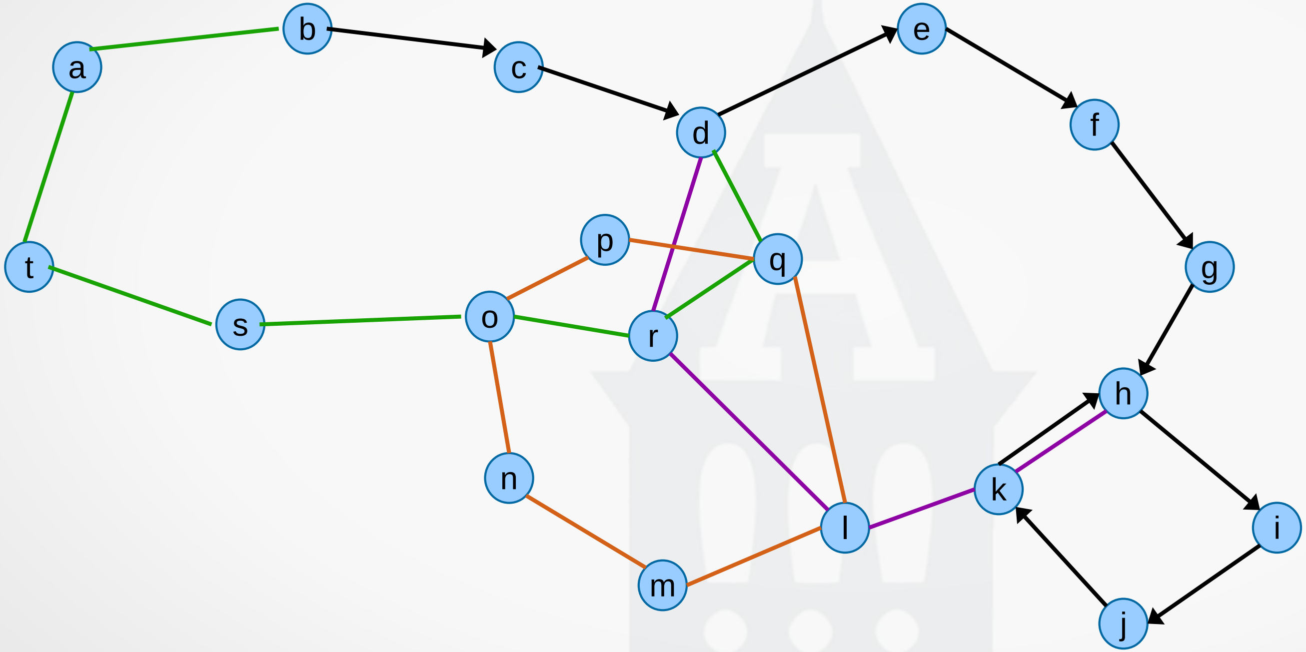
blue

purple

green

purple is not a new cycle, continue with blue

Euler Circuit – Algorithm : Joining Cycles

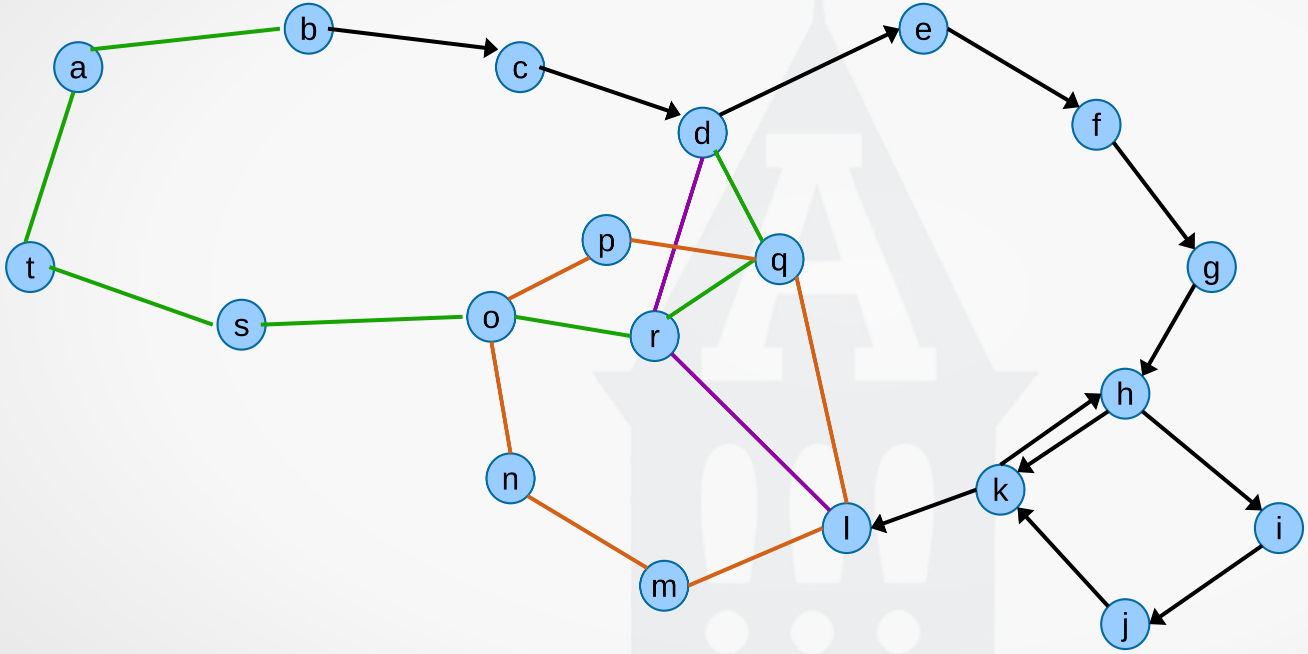


blue is finished, return to the purple cycle

purple

green

Euler Circuit – Algorithm : Joining Cycles

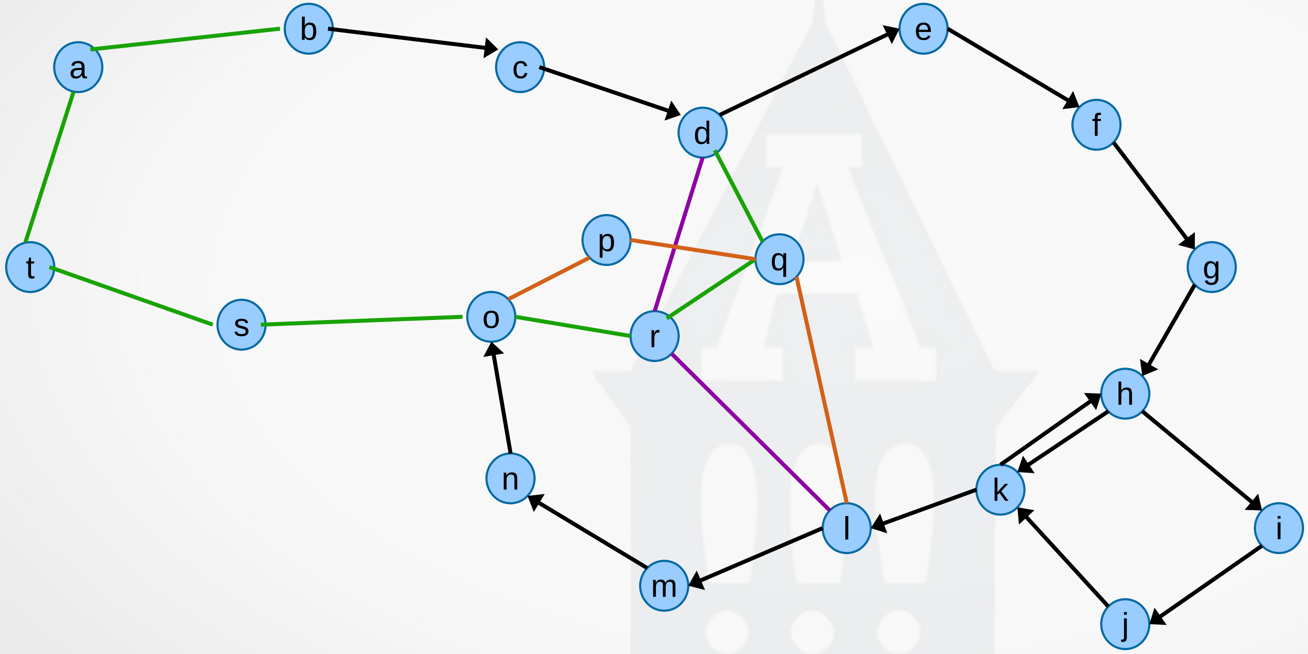


purple

green

continue along the purple cycle, encounter orange

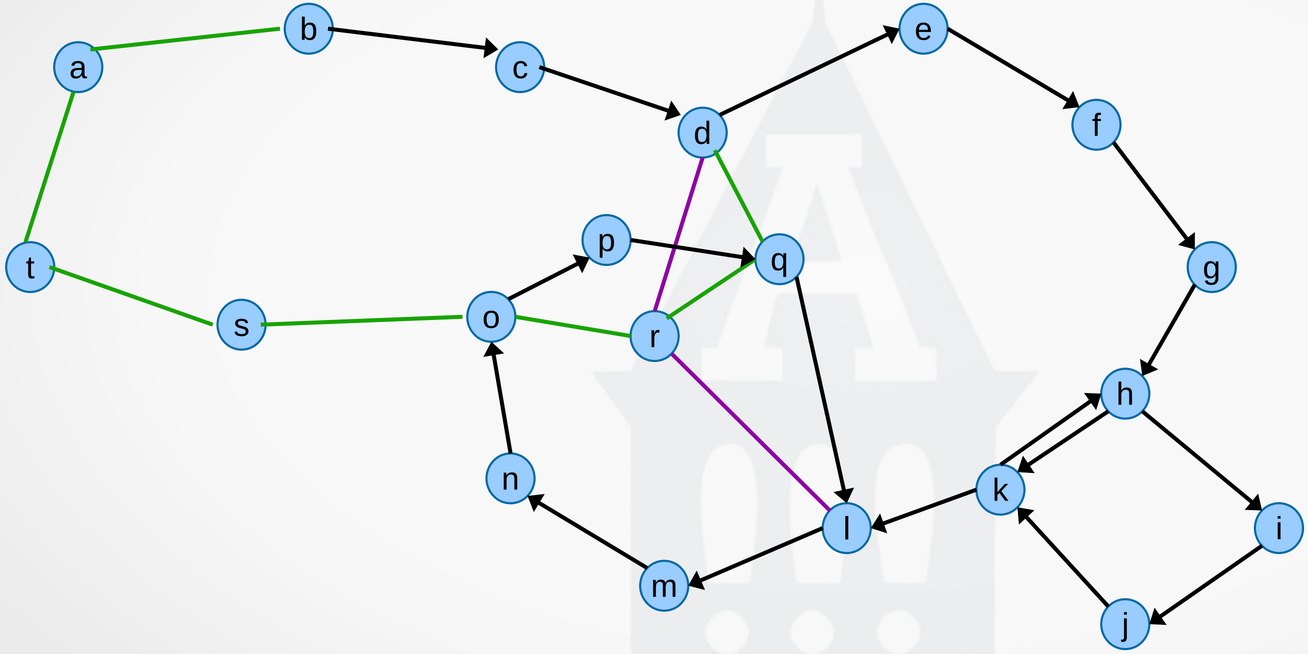
Euler Circuit – Algorithm : Joining Cycles



orange started, green is not a new cycle

orange
purple
green

Euler Circuit – Algorithm : Joining Cycles

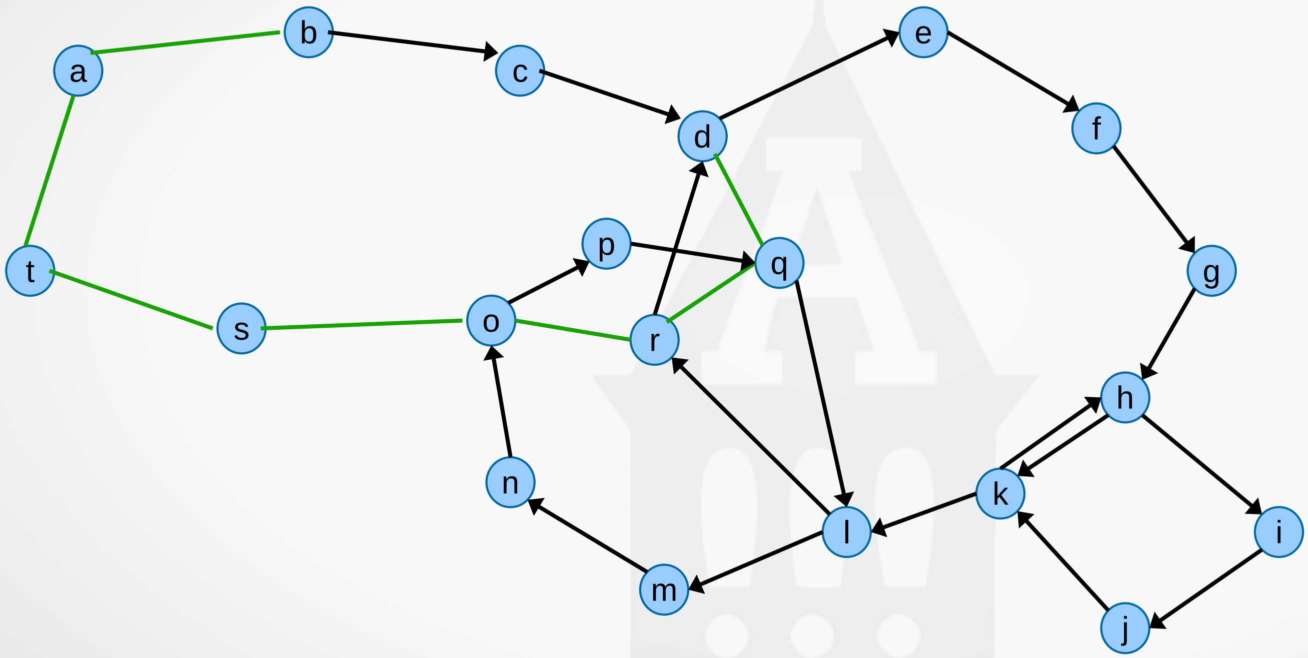


purple

green

finished orange, return to purple (again!)

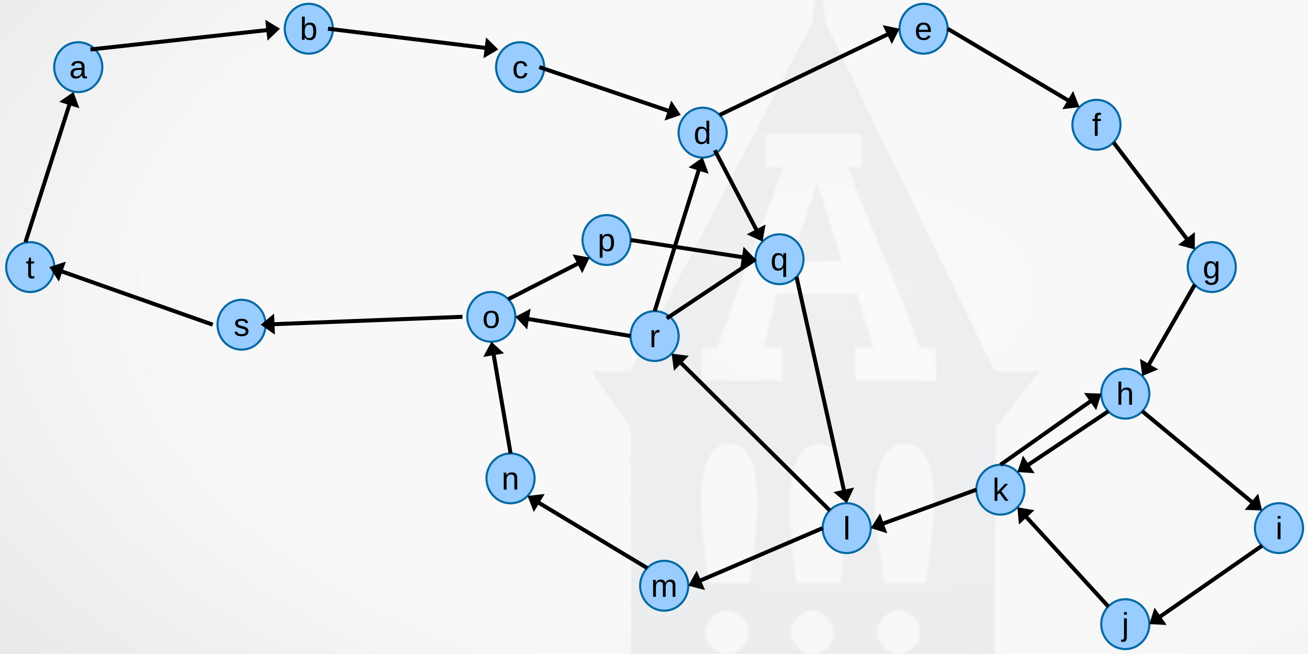
Euler Circuit – Algorithm : Joining Cycles



green

finished purple, return to green

Euler Circuit – Algorithm : Joining Cycles



all done!!

Hamiltonian Paths and Circuits

- A ***Hamilton path*** in a graph G is a path which visits every vertex in G exactly once
- A ***Hamilton circuit*** is a Hamilton path that returns to its start

- Difference is visiting every node/vertex versus traversing every edge

https://en.wikipedia.org/wiki/Icosian_game

Hamiltonian Paths and Circuits – Dodecahedron

- Find a path which visits all vertices?
- Find a circuit which visits all vertices?
- Can you draw it as a graph?

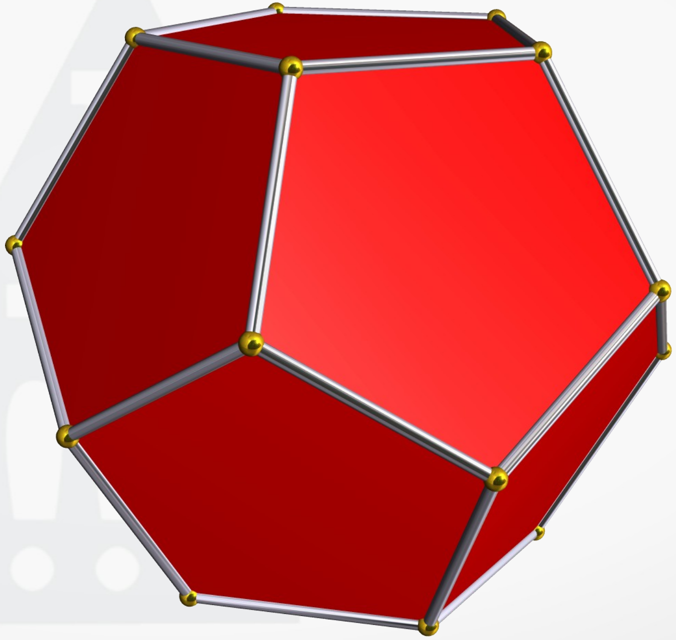


Image Credit: Robert Webb's Stella Software - http://www.software3d.com/Stella.php

Hamiltonian Paths and Circuits – Dodecahedron

- Find a path which visits all vertices?
- Find a ***circuit*** which visits all vertices?
- Can you draw it as a graph?

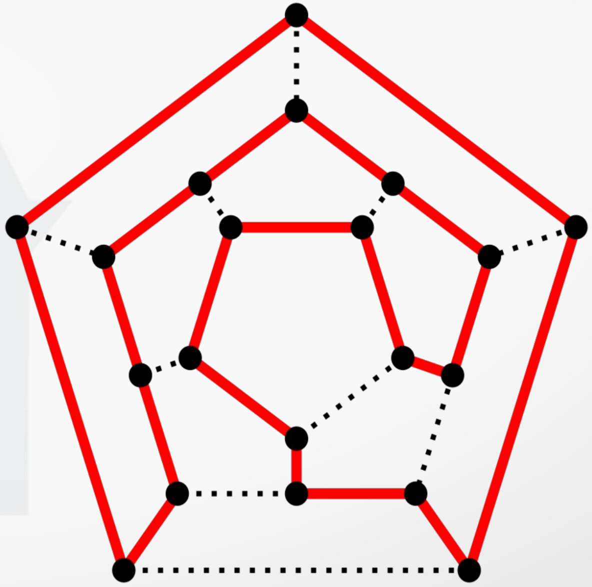
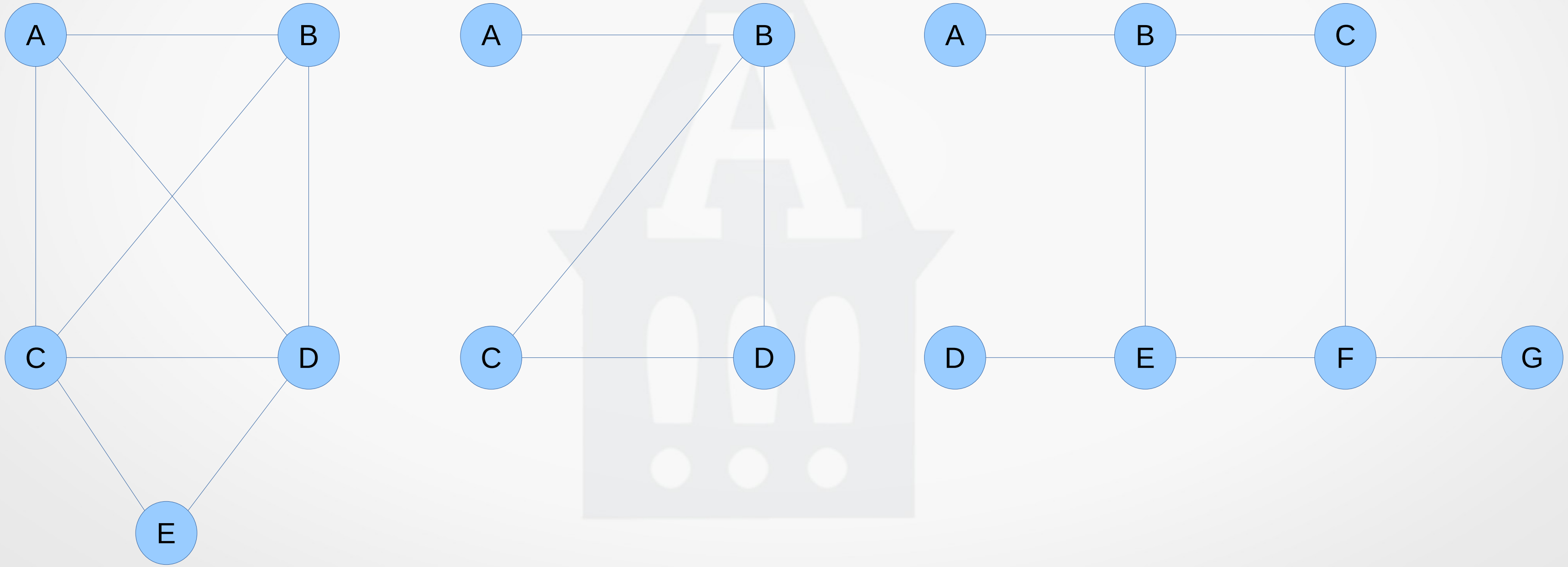


Image Credit: By Arne Nordmann, CC BY-SA 4.0, https://en.wikipedia.org/wiki/Icosian_game#/media/File:Hamiltonian_path_3d.svg

Image Credit: By Christoph Sommer - Own work, CC BY-SA 3.0, https://en.wikipedia.org/wiki/Icosian_game#/media/File:Hamiltonian_path.svg

Hamilton Paths and Circuits

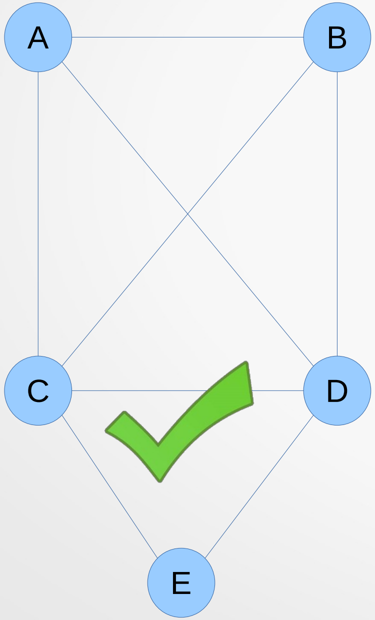
- Which of these graphs has a Hamilton ***path***?



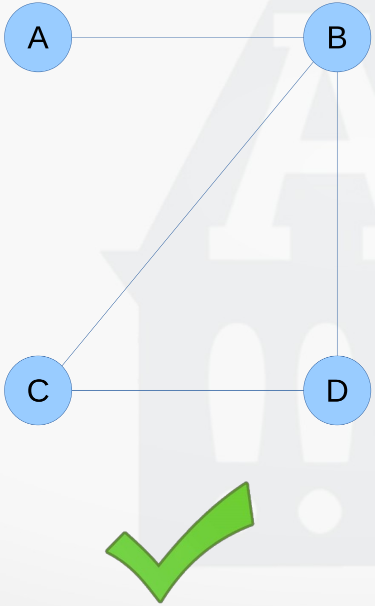
Hamilton Paths and Circuits

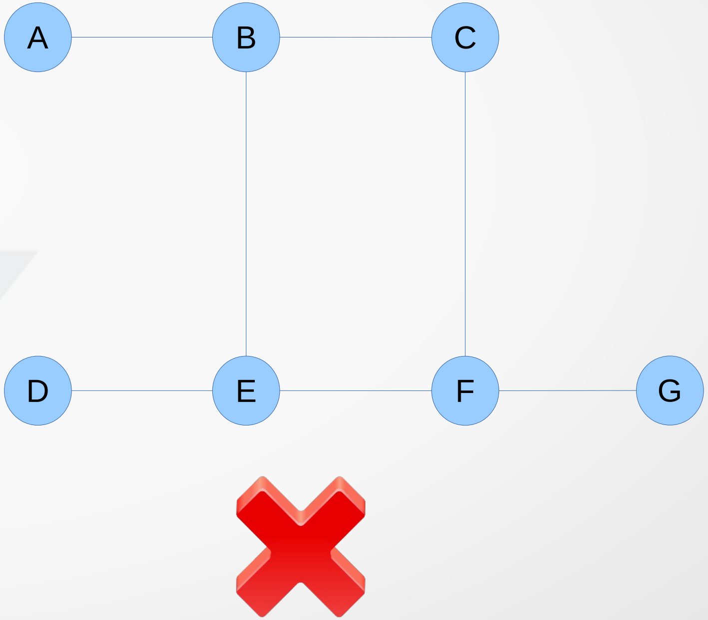
- Which of these graphs has a Hamilton ***path***?

a, b, c, d, e



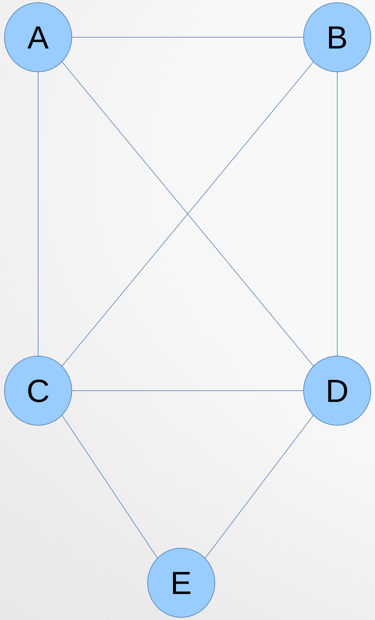
a, b, c, d

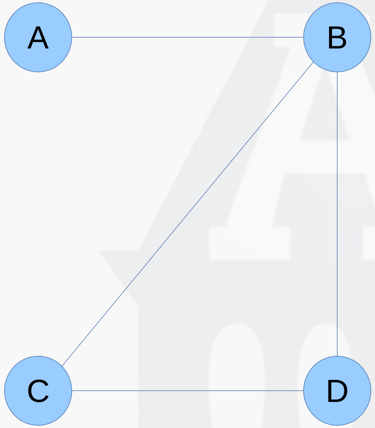


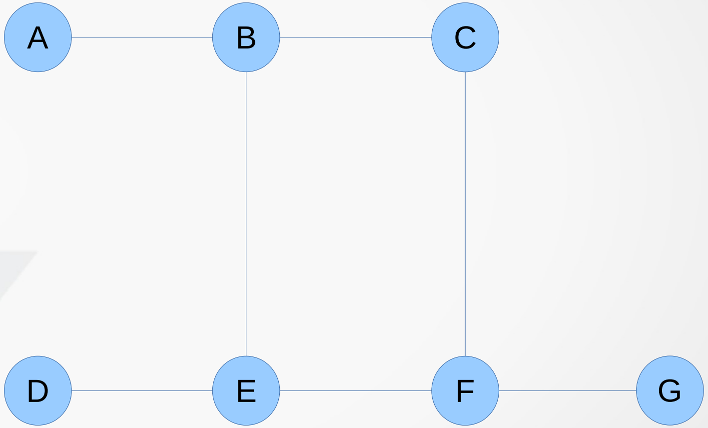


Hamilton Paths and Circuits

- Which of these graphs has a Hamilton ***circuit***?



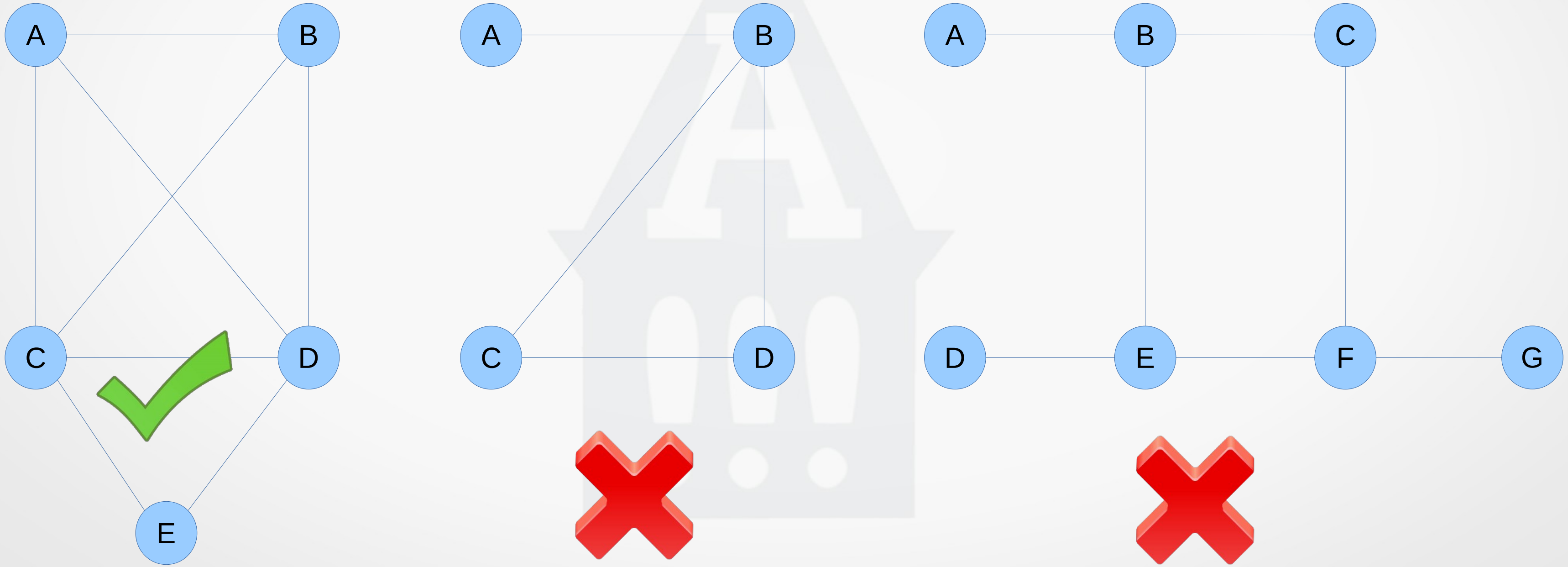




Hamilton Paths and Circuits

- Which of these graphs has a Hamilton ***circuit***?

a, b, d, e, c, a



Hamilton Paths and Circuits

- Unlike Euler circuit problem, finding Hamilton circuits is hard
- There is no simple set of necessary and sufficient conditions, and no simple algorithm

Hamilton Paths and Circuits – Applications

- Determining routes for things like…
  - Mail delivery
  - Food delivery
  - Garbage & Recycling pickup
  - Bus service
  - Traveling Salesperson Problem (TSP)
    - Find a Hamilton circuit in a complete graph such that the total weight of its edges is minimal
  - Anything that needs to visit all locations

Hamilton & Euler Paths and Circuits – Summary

Property	Euler	Hamilton
Repeated visits to a given node allowed	Yes	No
Repeated traversals of a given edge allowed	No	No
Omitted nodes allowed	No	No
Omitted edges allowed	No	Yes