Graphs

Peeking into Computer Science

Reading Assignment

- Mandatory: Chapter 3 – Sections 3.1 & 3.2

Graphs

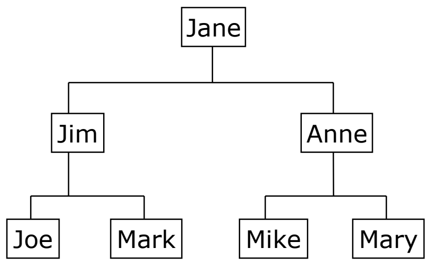
Abstraction of Data

At the end of this section, you will be able to:

1. Define directed and undirected graphs
2. Use graphs to model data
3. Use graph terminology
4. Represent graphs using adjacency matrices

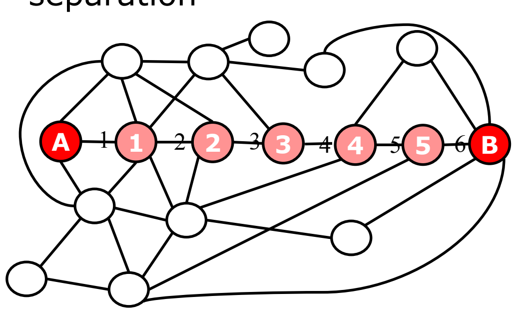
Objectives

- To show relationships (people, objects, locations etc.)



JT's Extra: What Is A Graph Used For?

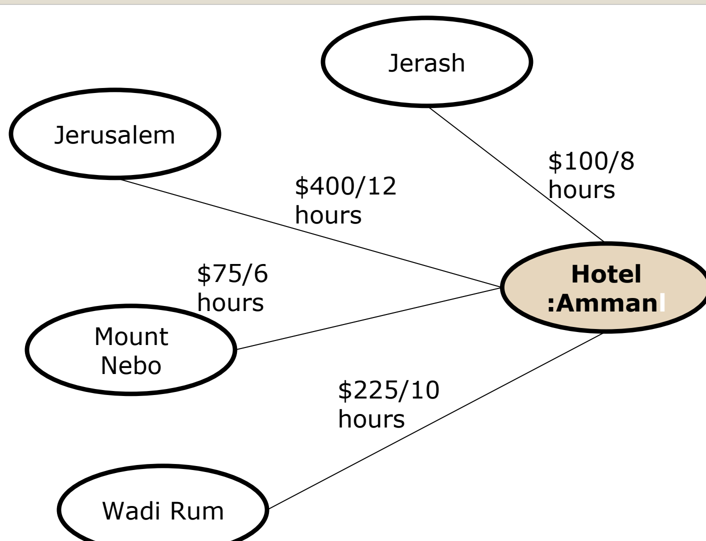
- Visualizing connections e.g., "6 degrees of separation"



JT's Extra: What Is A Graph Used For?

- They can be used anytime that the relationships between some entities must be visualized.
- Examples:
  - A few from will be used in the following slides:
    - http://www.graph-magics.com/practic_use.php

JT's Extra: Practical Applications (Graphs)



JT's Extra Example: JT's Vacation[1]

JT's Extra: Other Examples[1]

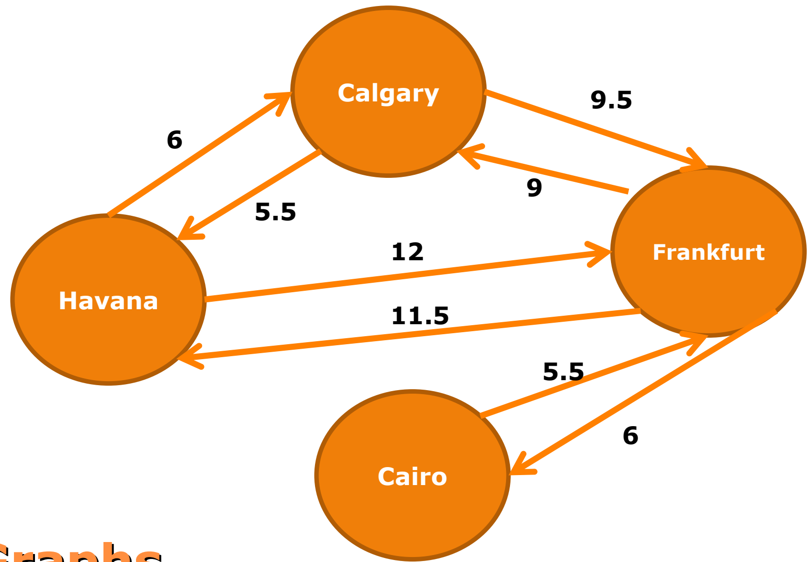
- Logistics:
  - Making sure that you deliver your items to every street within a part of the city. You want to cover every street but at the same time minimize travel time.
  - Telecommuincations: Find the cheapest way to connect communication stations (TV, telephone, computer network) so that a station is connected to any other (directly, or through intermediate stations).

1 http://www.graph-magics.com/practic_use.php

JT's Extra: Other Examples[1] (2)

- (More logistics)
  - A warehouse should be placed in a city (a region) so that the sum of shortest distances to all other points (regions) is minimal. This is useful for lowering the cost of transporting goods from a warehouse to clients.

1 http://www.graph-magics.com/practic_use.php

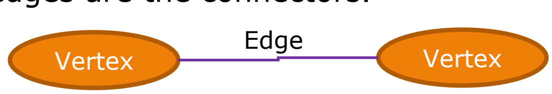


Graphs

- A graph G is defined as:
  G = (V,E)
  where

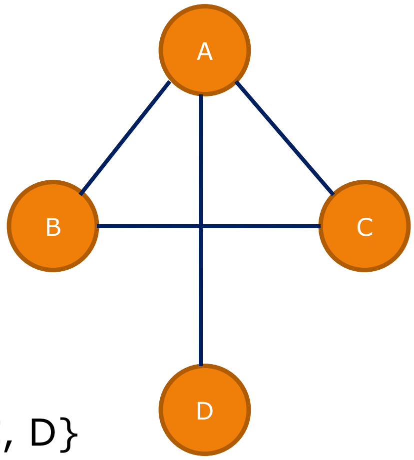
  V(G) is a set of vertices
  E(G) is a set of edges (pairs of vertices)

  JT's note: Vertices are 'things' being connected, edges are the connectors.



- Directed Graph: edges are one-way
- Undirected Graph: edges are two-way
- Labeled Graphs: edges have weights

Definition

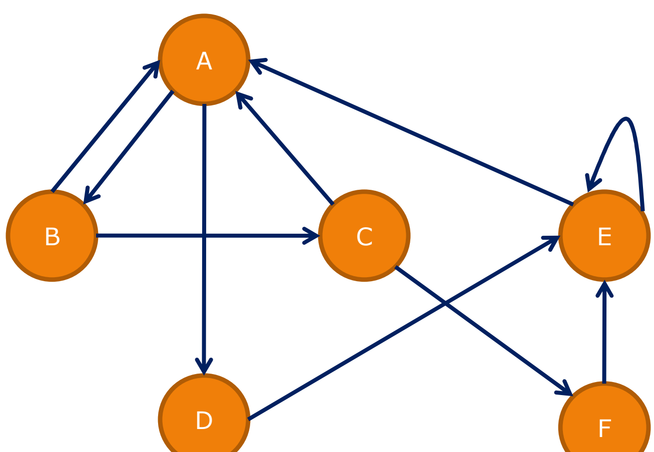


- V = {A, B, C, D}
- E = {{A,B},{A,C},{A,D},{B,C}}

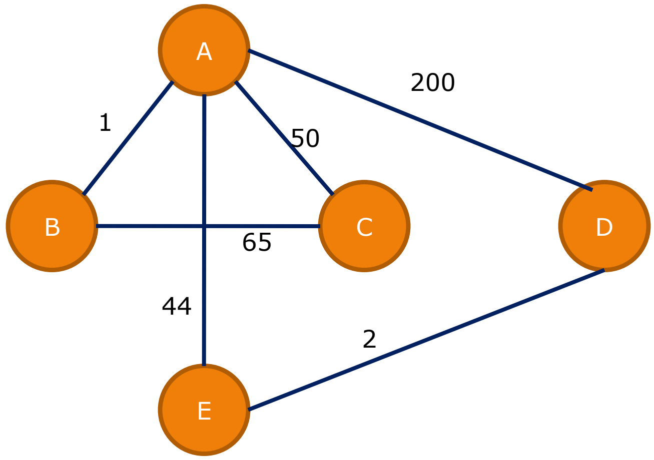
An undirected graph

V = {A, B, C, D, E, F}

E = {
(A,B),
(B,A),
(B,C),
(A,D),
(C,A),
(D,E),
(C,F),
(E,A),
(E,E),
(F,E)}



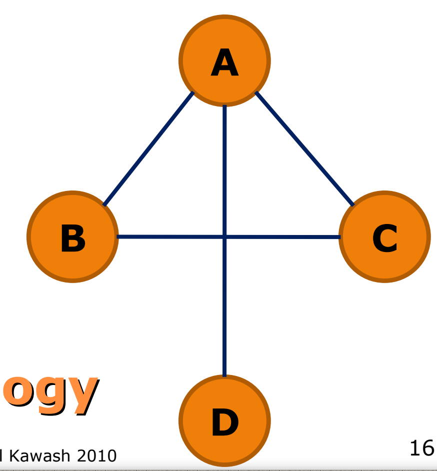
A directed graph



A labeled undirected graph

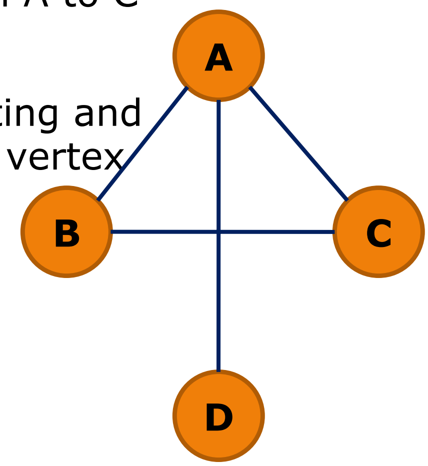
Graph Terminology

- **Adjacent** vertices: connected by an edge
- A and B are adjacent
- D and C are not



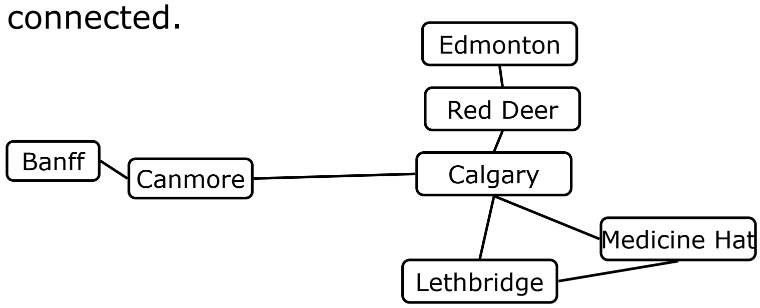
Graph Terminology

- There is a **Path** from A to C
  - One path: A to B to C
- **Cycle** = a path starting and ending with the same vertex



JT's Extra: Path Example

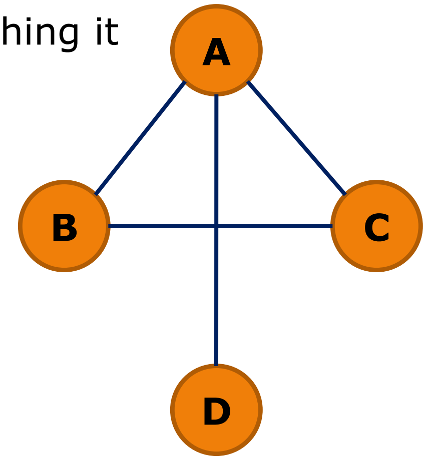
- There is a path between vertices if they are directly connected by an edge or indirectly connected.



  - Red Deer and Lethbridge are not adjacent but there is at least one path from Red Deer to Lethbridge.

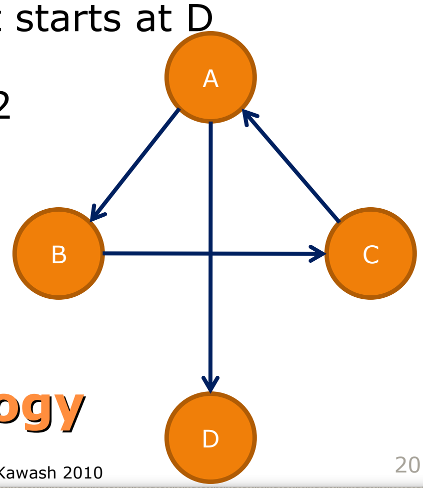
Graph Terminology

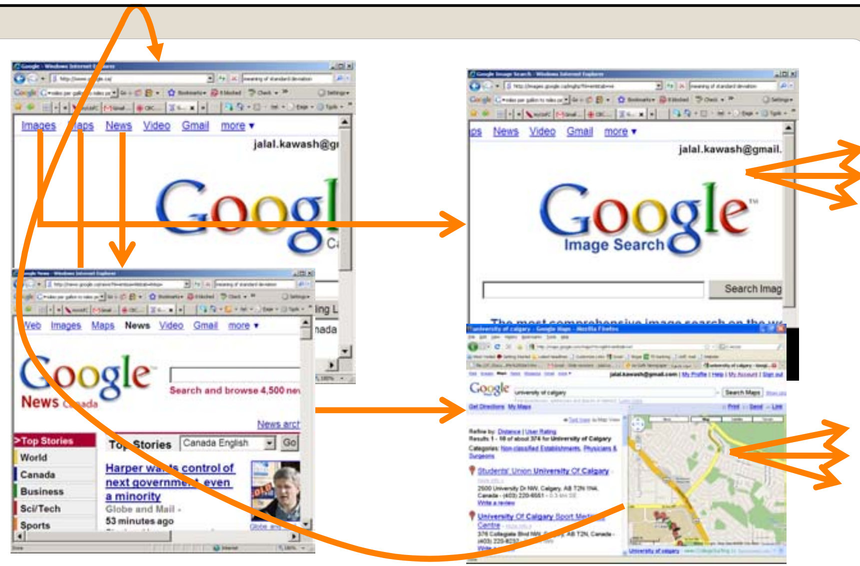
- **Degree** of a vertex = Number of edges touching it
  degree of C = 2
  degree of A = 3



Graph Terminology

- A is adjacent **to** B
- B is adjacent **from** A
- There is a **Path** from A to C
- There is no path that starts at D
- **In-degree** of A is 1
- **Out-degree** of A is 2



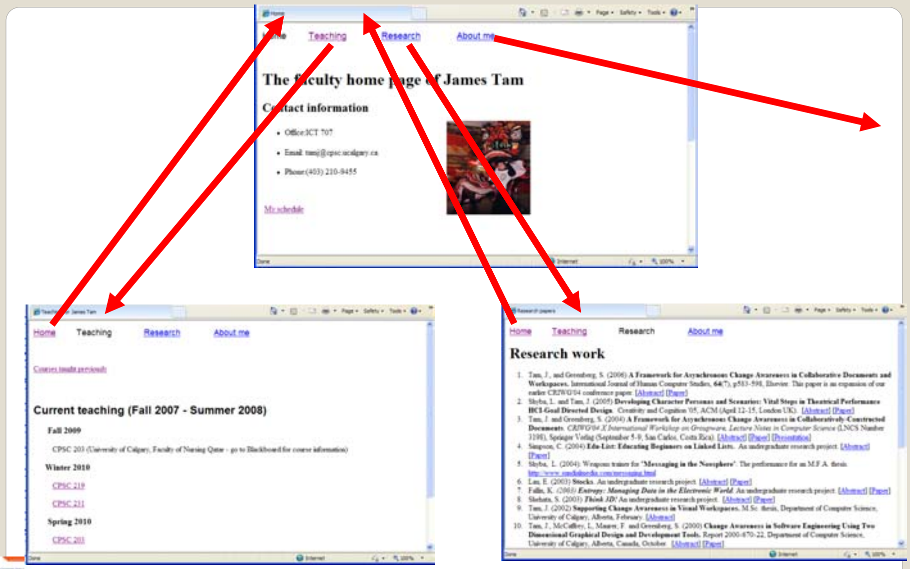


The Web as a Directed Graph

- The World Wide Web itself can be visualized as a directed graph.
  - Vertex = web page, Edge = link between pages.

The WWW Is A Graph

Peeking into Computer Science © Jalal Kawash 2010

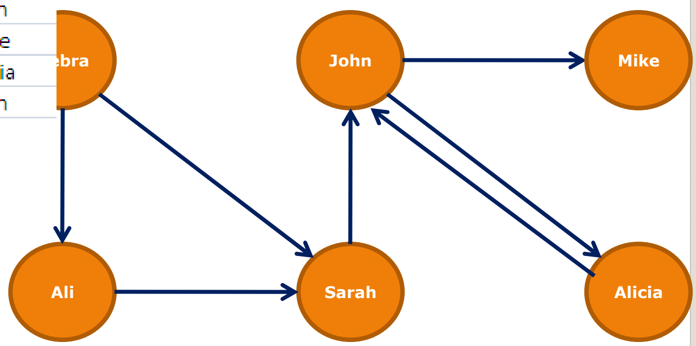


The WWW Is A Graph (2)

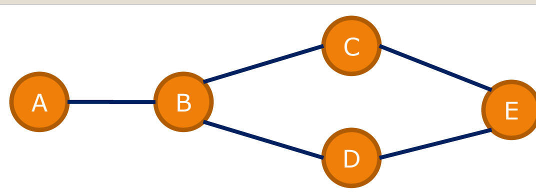
- Visualizing the layout of a page as a graph can be useful in web design.
  - Are there sufficient connections between pages?
  - Do the connections (links) make sense to visitor?
  - Although it should be possible to reach any page without too much clicking (rule of thumb is 3), there are some pages that should always be accessible e.g., home page, contact page etc.

JT's Extra: Applying Graphs To Web Design

	A	B
1	Debra	Ali
2	Debra	Sarah
3	Ali	Sarah
4	Sarah	John
5	John	Mike
6	John	Alicia
7	Alicia	John



Graph Representation

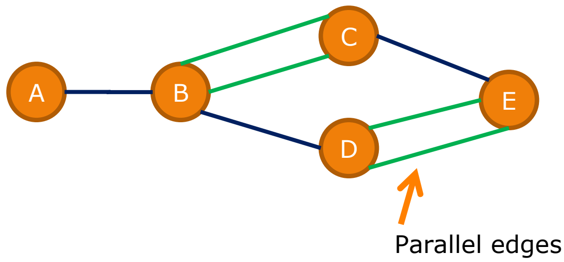


	A	B	C	D	E
A	0	1	0	0	0
B	1	0	1	1	0
C	0	1	0	0	1
D	0	1	0	0	1
E	0	0	1	1	0

Graph Representation

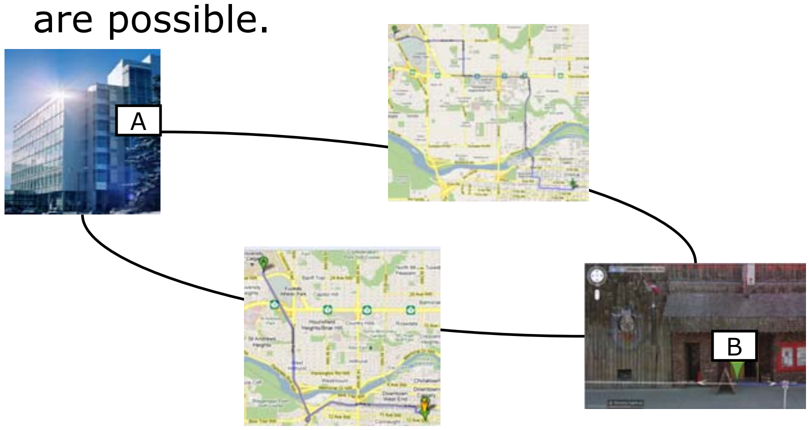
Multigraphs

- In a multigraph the set of edges is a multiset
- Edges can be repeated
- Graphs are special cases of multigraphs



JT’s Extra: Application Multigraphs

- Example: path finding when alternatives are possible.



Euler Paths

At the end of this section, you will be able to:

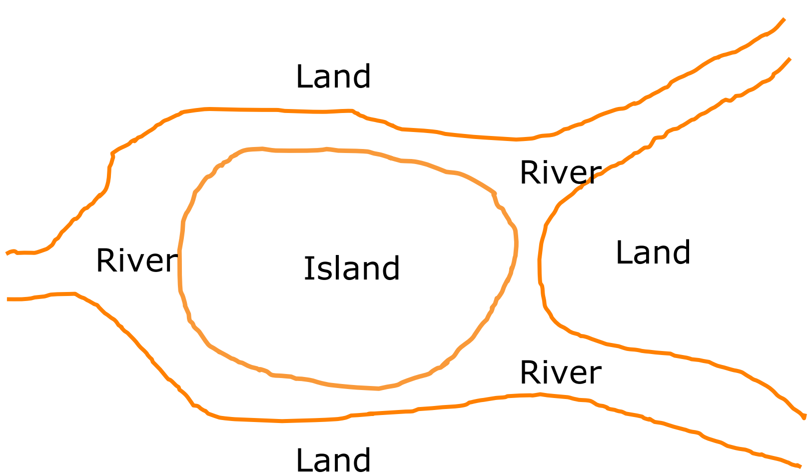
1. Understand how graphs are used to represent data
2. Appreciate the ability of graphs to lead to generalized conclusions
3. Define Euler tours and paths
4. Identify under which conditions an Euler circuit or path exists in a graph
5. Understand why such conditions are required
6. Learn the Euler circuit algorithm

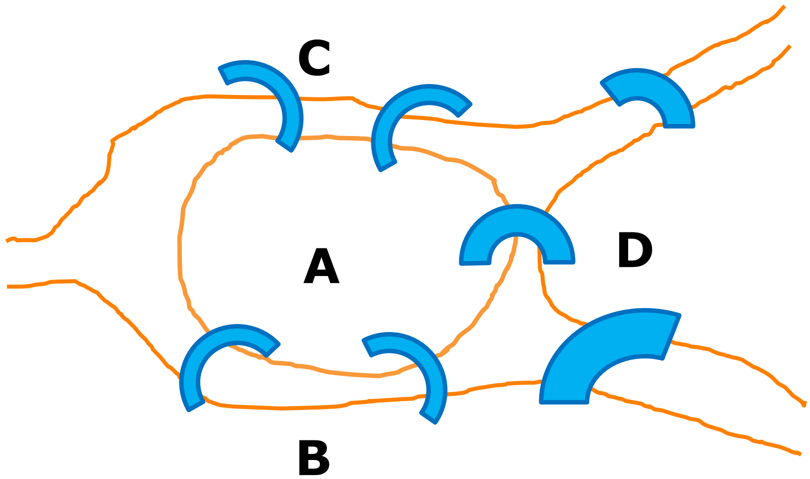
Objectives

Konigsberg

- Today called Kaliningrad in the Russian Republic
- In 1736 was in Prussia
- The town had seven bridges on the Pregel River

Konigsberg Layout





Seven Bridges

- People in 1736 wondered if it is possible to take the following walk in town:

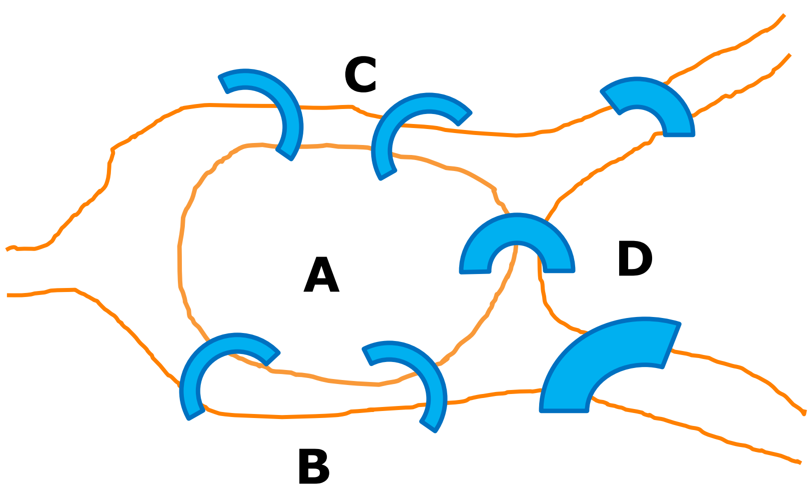
- Start at some location in town
- Travel across all bridges without crossing any bridge twice
- End the walk where it started

- They wrote to Swiss Mathematician Leonhard Euler for help

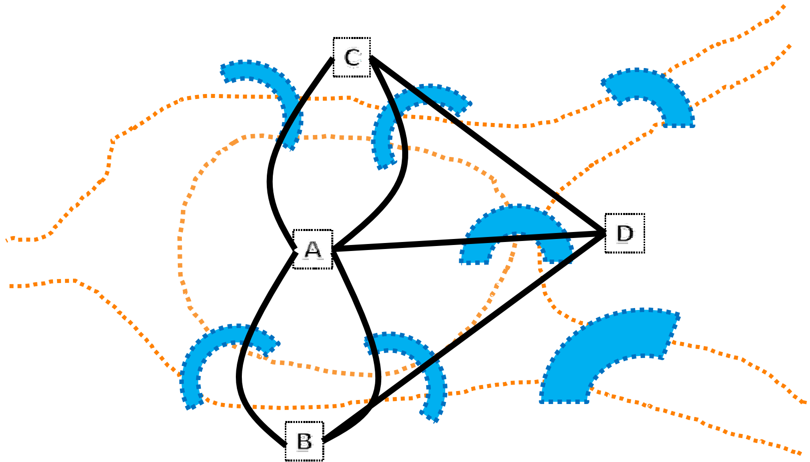
Konigsberg’s Walk

- 1707 –1783
- The greatest mathematician of the 18th century and one of the greatest of all time
- a pioneering Swiss mathematician and physicist
- important discoveries in graph theory
- introduced much of the modern mathematical terminology and notation

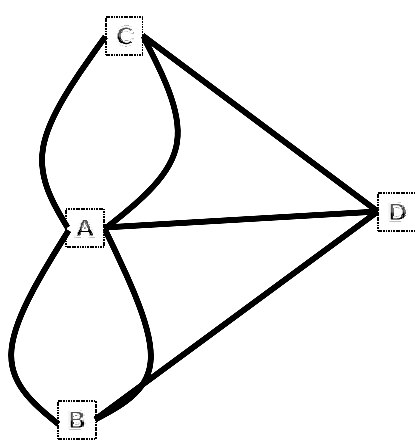
Leonhard Euler



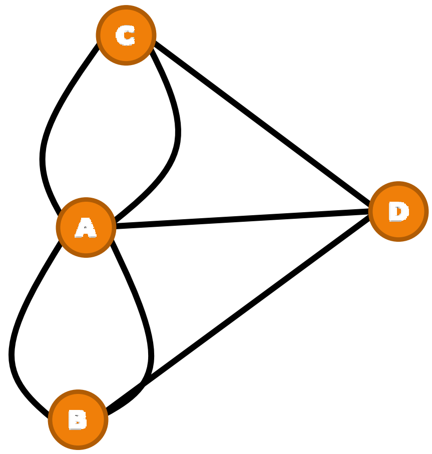
Konigsberg Multigraph



Konigsberg multigraph



Konigsberg multigraph



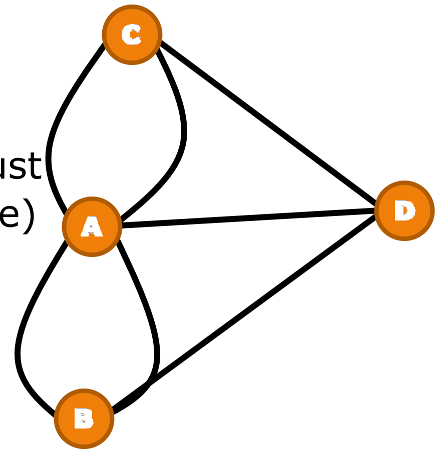
Konigsberg multigraph

- Is there a path through this graph that
  - Starts at a vertex
  - Ends in the same vertex
  - Passes through every edge once
  - Does not cross an edge more than once?
- Called an **Euler Circuit**
- Such a walk is impossible on any graph as long as the graph has one vertex with an *odd* degree

Euler's Response

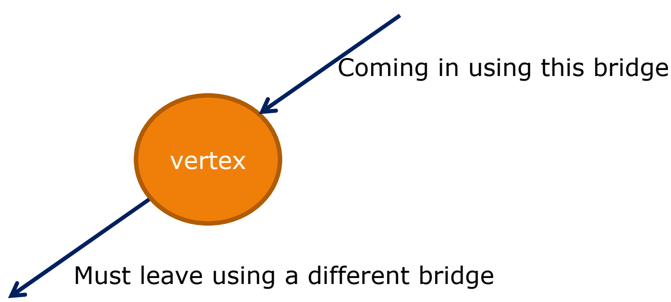
Euler's Response

- There is at least one vertex of an odd degree
- Does not work
(the start vertex must have an even degree)



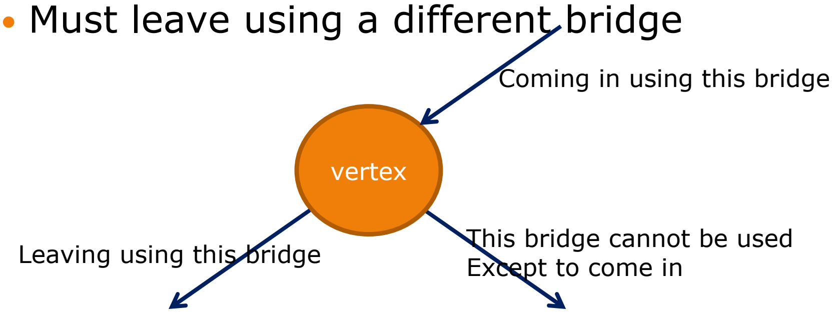
Even-Degree Vertex

- To cross every bridge once



- Each vertex must have an even degree

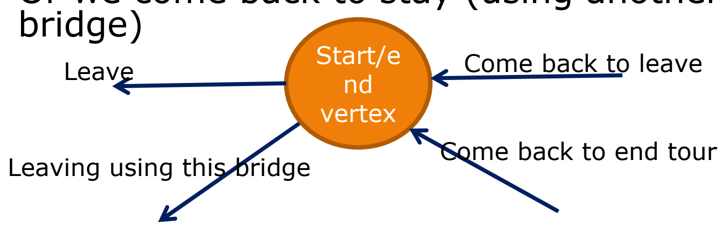
- Must come in using some bridge
- Must leave using a different bridge



- We're stuck

Odd-Degree Vertex

- Must leave using some bridge
- Either we come back to leave again (need two new bridges)
- Or we come back to stay (using another bridge)



- Cannot be of an odd degree: we need another bridge to come back

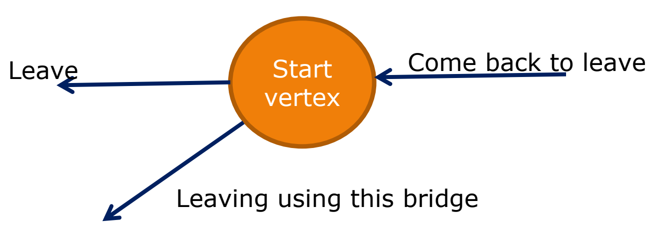
The Starting/Ending Vertex

An Easier Problem

- Is there a path through this graph that
  - Starts at some vertex
  - Ends at a (possibly **different**) vertex
  - Passes through every edge once
  - Does not cross an edge more than once?
- Called a **Euler Path**
- Note that if there is a circuit, then there is a path
- If not, then the path is necessarily not a circuit

Euler Path – Start Vertex

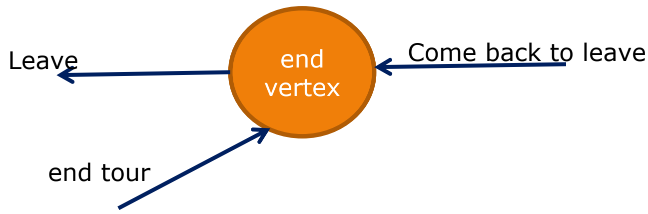
- The starting vertex must have an odd degree



- **Never** come back to stay

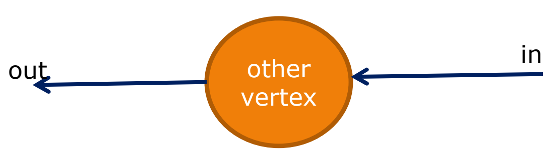
Euler Path – End Vertex

- The end vertex must have an odd degree



Other Vertices

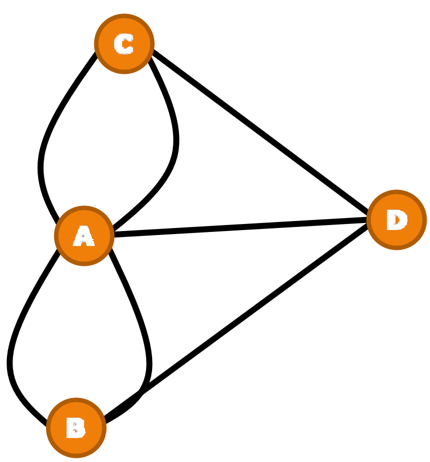
- All other vertices must have an even degree



- Every **in** must have a matching **out** on "new" bridges

- Start and end vertices must have odd degrees
- Every other vertex must have an even degree
- There is no Euler Path for the Konigsberg graph

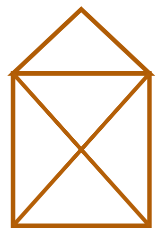
Euler (non-cycle) Path Requirements



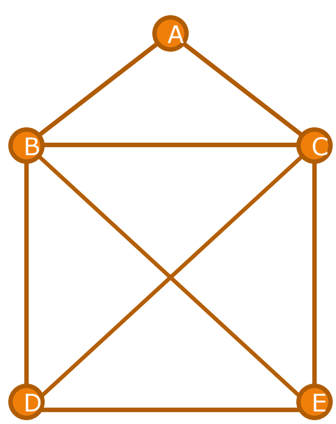
No Euler Path for this Graph

A Similar Problem

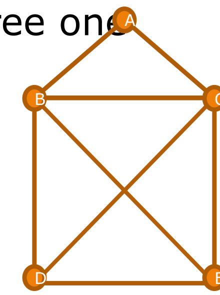
- Can you draw this shape with the rules:
  - Draw **continuously**, cannot lift pen from one position to another
  - Draw each line **once**, cannot let pen run on top of an already drawn line



Graph Representation



- Is there a Euler Circuit
  - No (some vertices have odd degrees)
- Is there a Euler Path
  - Yes (exactly two vertices have odd degrees)
- Path must start at an odd-degree vertex and ends at another odd-degree one

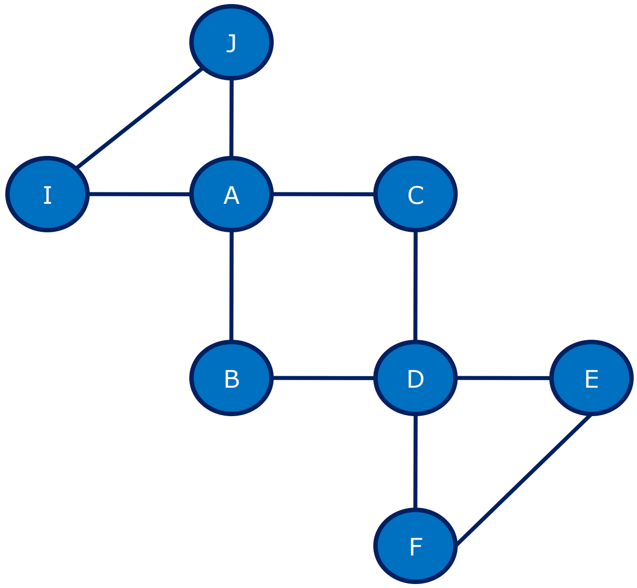


Graph Representation

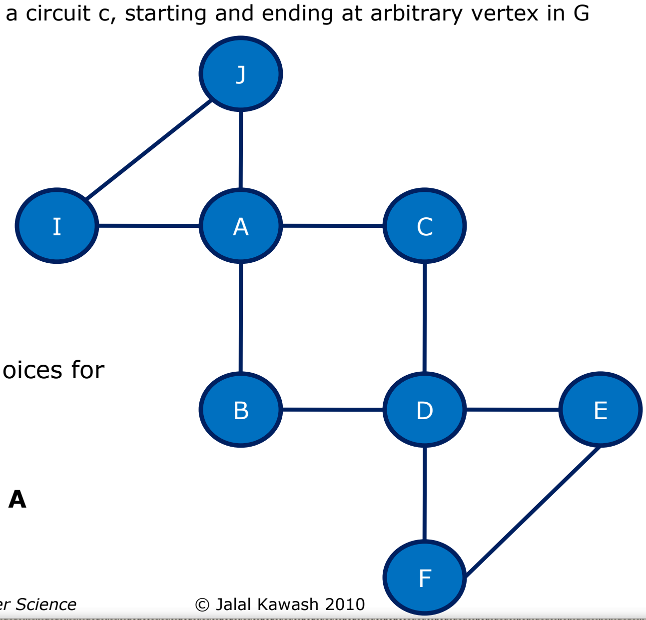
Given a graph G (all vertices have even degrees):

1. Construct a circuit c, starting and ending at arbitrary vertex in G
2. Remove all the edges of c from G
3. Repeat until G has no edges:
   a) Construct a circuit c’ in G that starts (ends) in a vertex v that is in c
   b) Add c’ to c at v
   c) Remove all the edges of c’ from G

Euler Circuit Algorithm



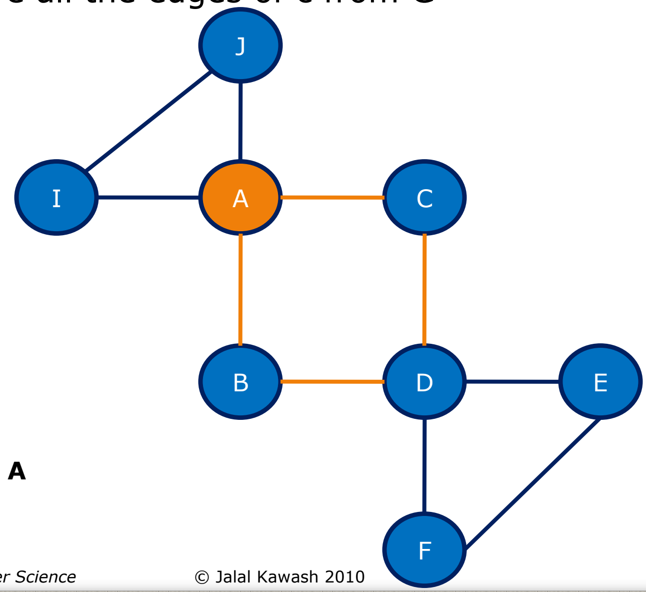
1. Construct a circuit c, starting and ending at arbitrary vertex in G



Note: other choices for c are possible

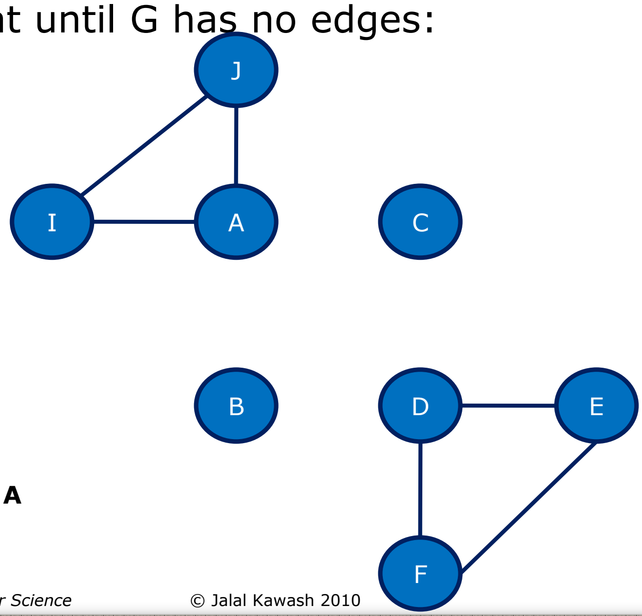
c: A, B, D, C, A

2. Remove all the edges of c from G



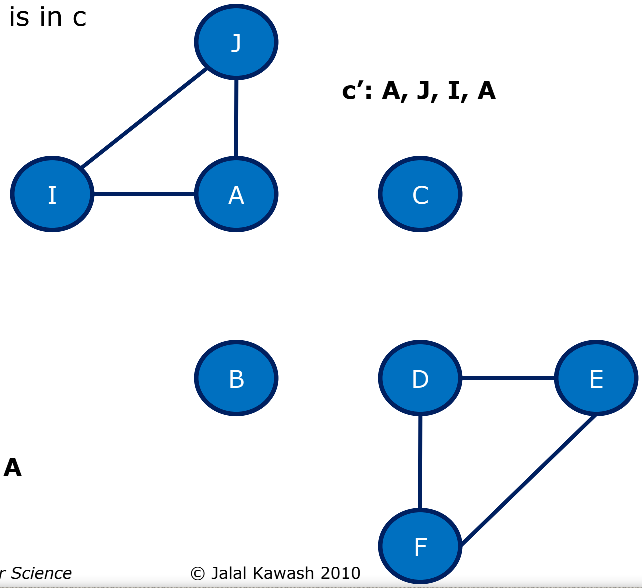
c: A, B, D, C, A

3. Repeat until G has no edges:



c: A, B, D, C, A

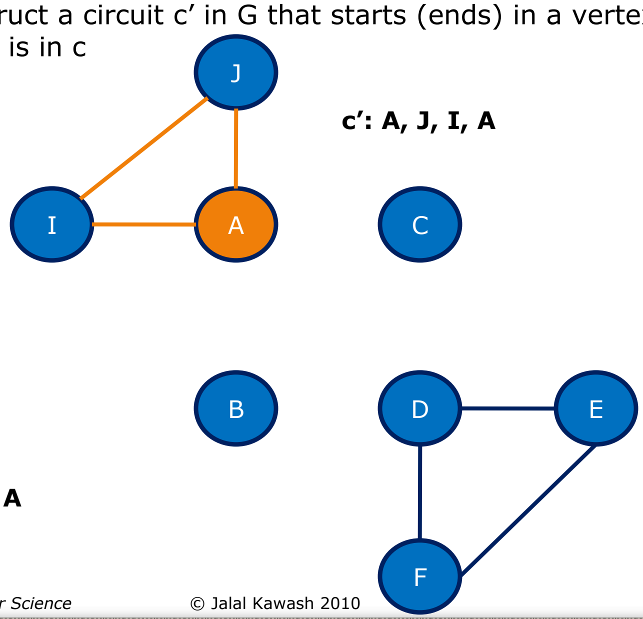
a) Construct a circuit c’ in G that starts (ends) in a vertex v that is in c



c’: A, J, I, A

c: A, B, D, C, A

a) Construct a circuit c’ in G that starts (ends) in a vertex v that is in c

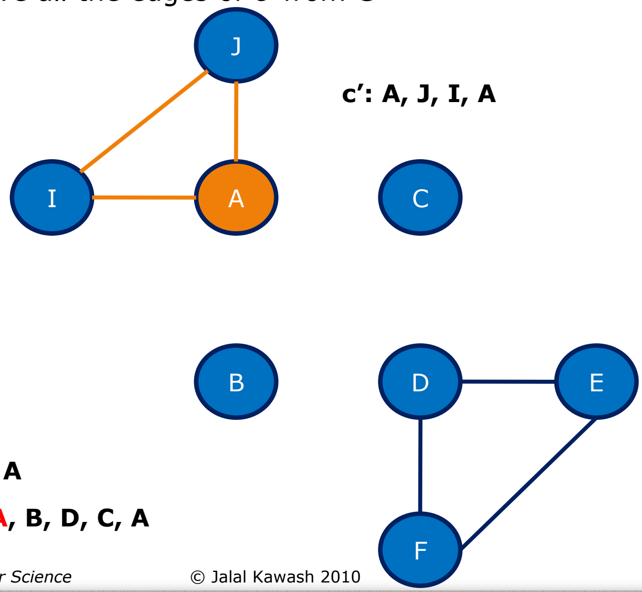


c’: A, J, I, A

c: A, B, D, C, A

a) Add c’ to c at v

b) Remove all the edges of c’ from G



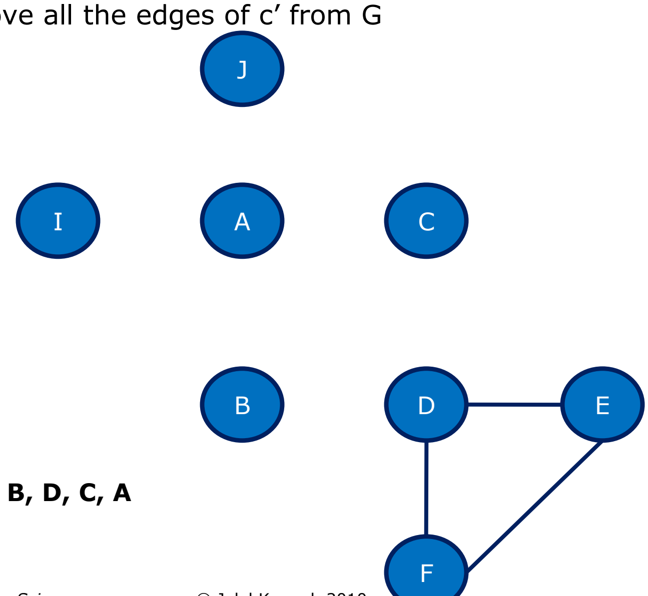
c’: A, J, I, A

c: A, B, D, C, A

c: A, J, I, A, B, D, C, A

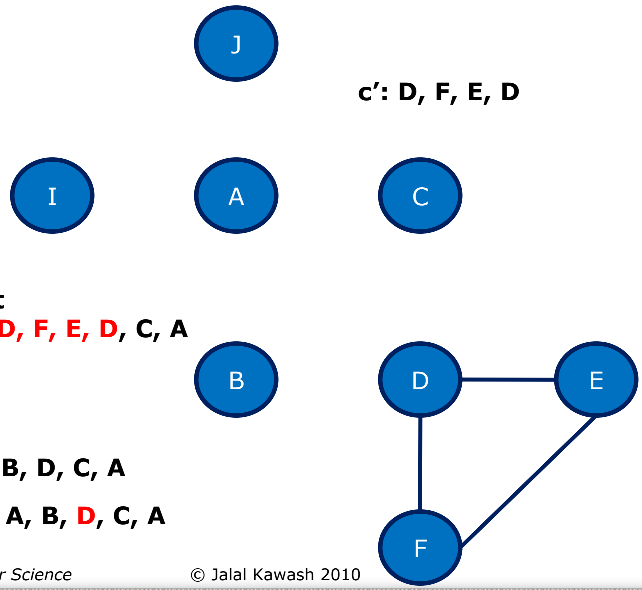
a) Add c’ to c at v

b) Remove all the edges of c’ from G



c: A, J, I, A, B, D, C, A

Repeat a), b), and c)



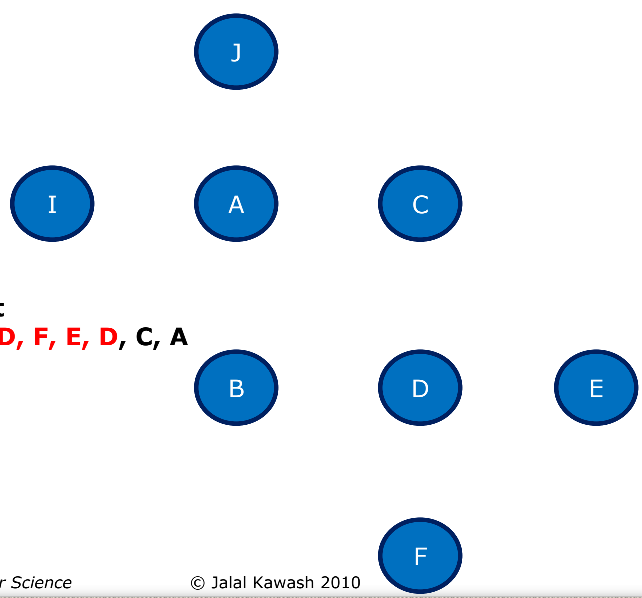
c′: D, F, E, D

An Euler circuit
c: A, J, I, A, B, D, F, E, D, C, A

c: A, J, I, A, B, D, C, A

c: A, J, I, A, B, D, C, A

The graph has no more edges, stop



An Euler circuit
c: A, J, I, A, B, D, F, E, D, C, A