Graphs And Algorithms

Peeking into Computer Science

- Mandatory: Chapter 3 – Sections 3.1 & 3.2

Reading Assignment

Graphs

Abstraction of Data

3

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

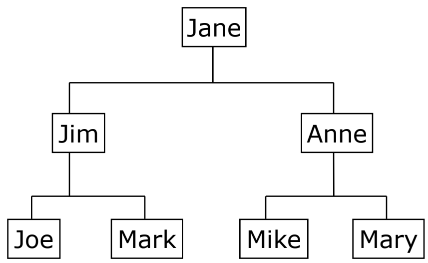
1. Define directed and undirected graphs
2. Use graphs to model data
3. Use graph terminology
4. Represent graphs using adjacency matrices

Objectives

Peeking into Computer Science © Jalal Kawash 2010

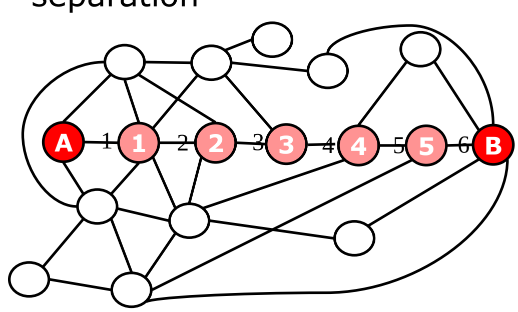
4

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



JT’s Extra: What Is A Graph Used For?

- Visualizing connections e.g., “6 degrees of separation”

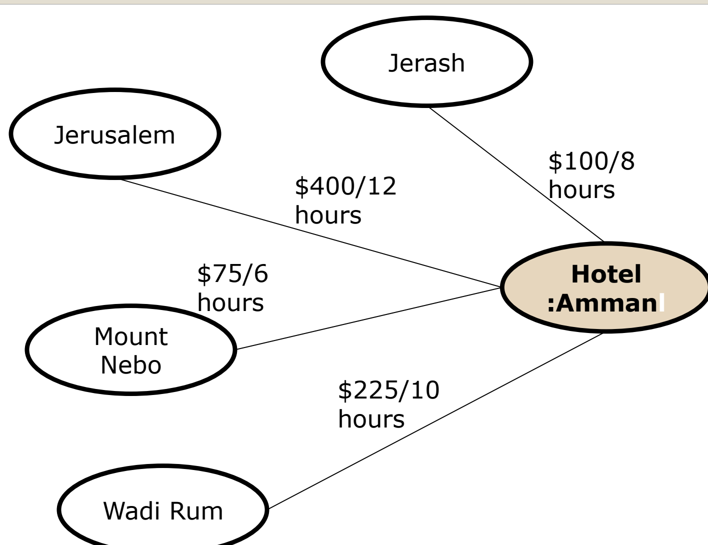


JT’s Extra: What Is A Graph Used For?

JT's Extra: Practical Applications (Graphs)

- 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 Example: JT's Vacation



- 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).

JT's Extra: Other Examples[1]

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

- (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.

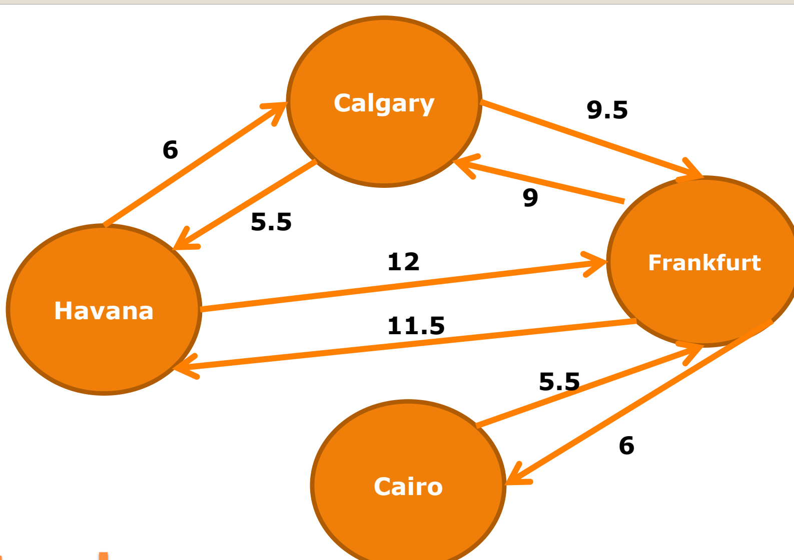
JT's Extra: Other Examples[1] (2)

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

JT's Extra: Why Learn Graph Theory?

- Solving everyday problems by abstracting details:
  - In this case the problem is represented by (abstracted into) a graph.
  - Other types of problems can be abstracted into other structures such as trees (later section).
- Removing extraneous details through abstraction may make the problem easier to solve (or even make the seemingly impossible possible).

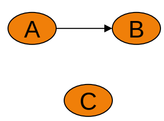
Graphs



JT's Extra: How Will Graphs Be Used In This Section

- To graphically show the relations in a set
- Example graph:
  - S = {A, B, C}

- A graph can show the relationships between elements

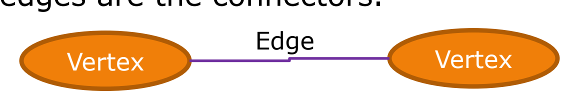


Definition

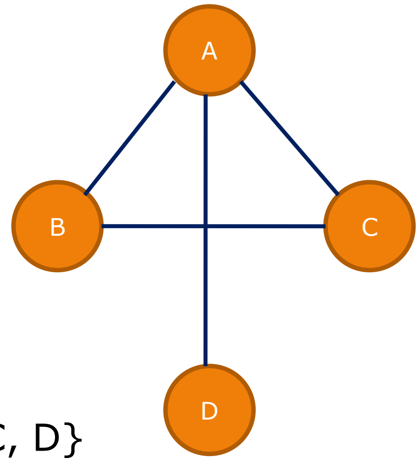
- A graph G is defined as:
  G = (V,E)
  where

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

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



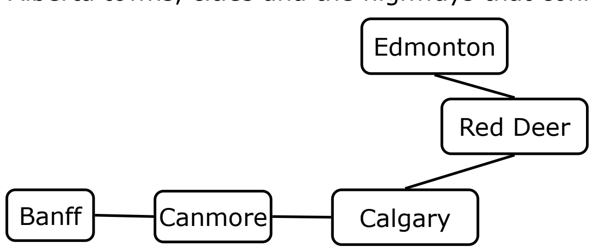
- Directed Graph: edges are one-way
- Undirected Graph: edges are two-way
- Labeled Graphs: edges have weights



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

An undirected graph

- Edges are two-way
  - Alberta towns, cities and the highways that connect them.

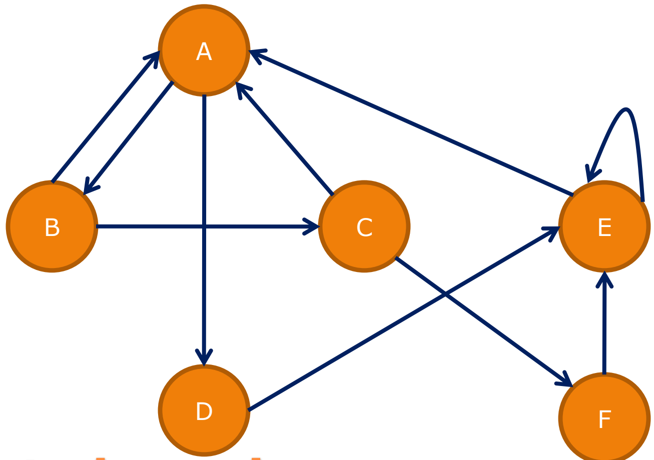


V = {Edmonton, Red Deer, Calgary, Canmore, Banff}
E = {{Banff, Canmore},
{Calgary, Canmore},
{Calgary, Red Deer},
{Edmonton, Red Deer}}

JT' Extra: Undirected Graphs

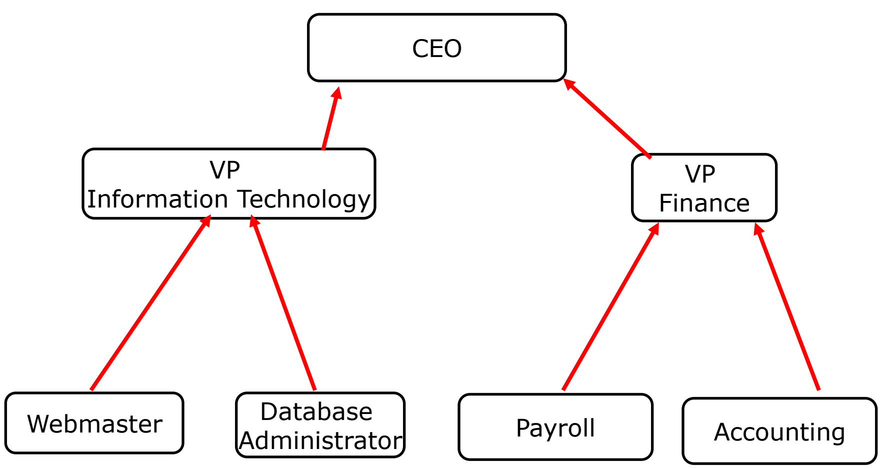
V = {A, B, C, D, E, F}

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

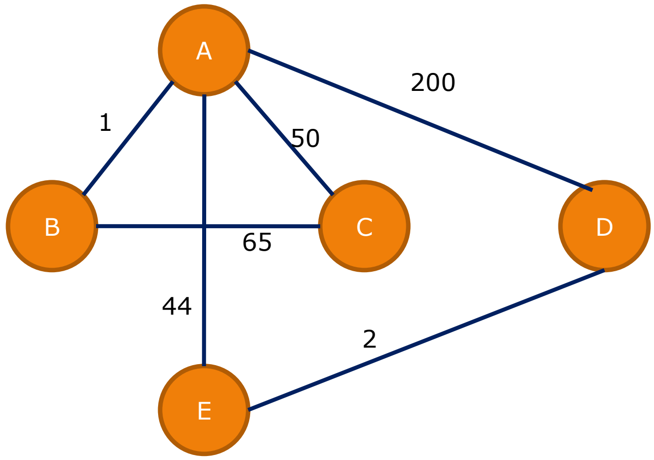


A directed graph

- Organizational structure ("reports to")

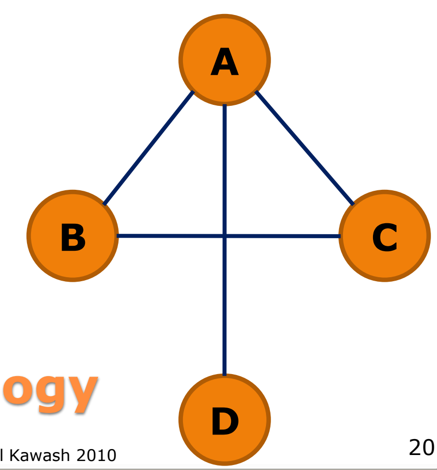


JT's Extra: Directed Graph



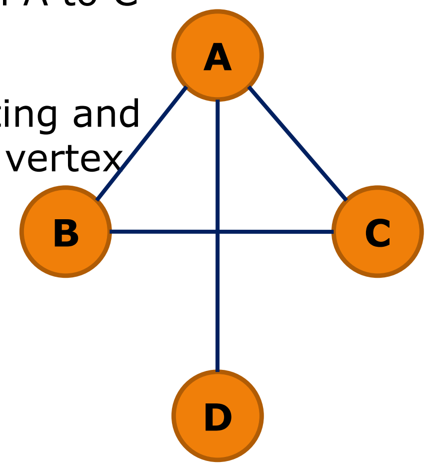
A labeled undirected graph

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



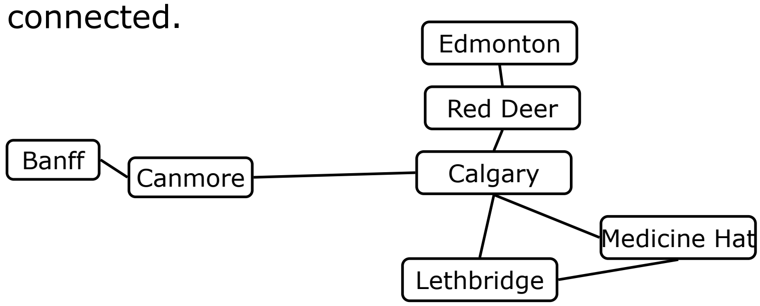
Graph Terminology

- 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



Graph Terminology

- There is a path between vertices if they are: 1) directly connected by an edge or 2) indirectly connected.

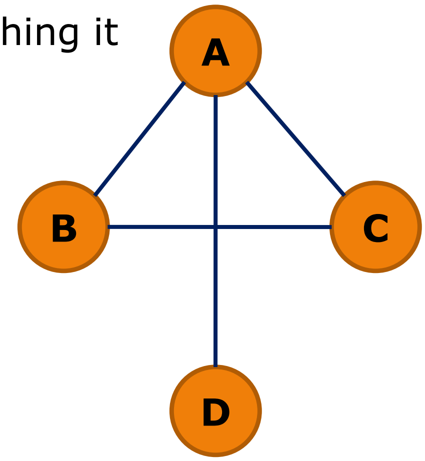


  - Red Deer and Lethbridge are not adjacent but there is at least one path from Red Deer to Lethbridge.

JT's Extra: Path Example

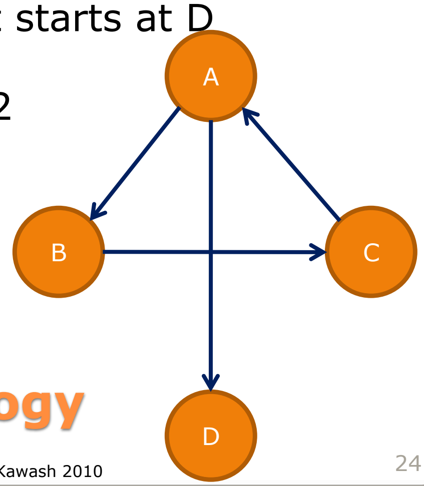
Graph Terminology

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



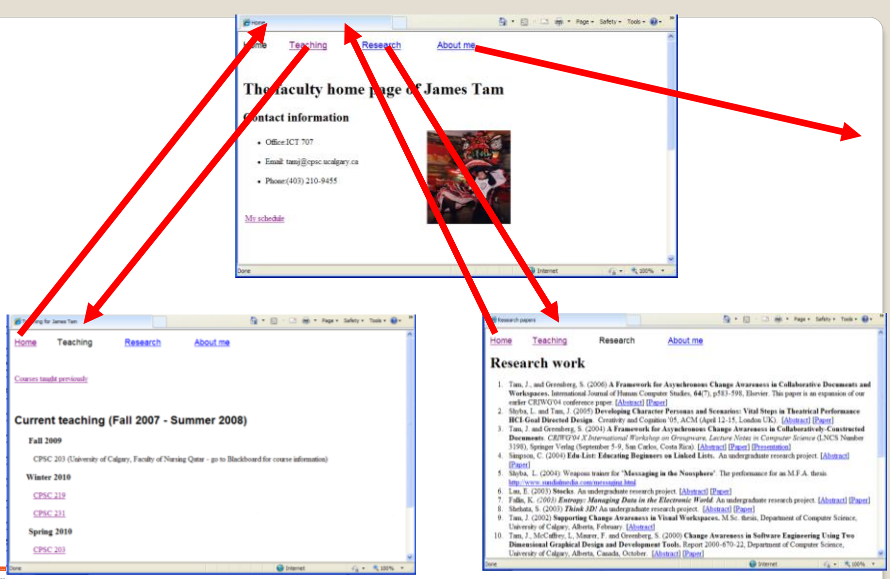
Graph Terminology

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

The WWW Is A Graph



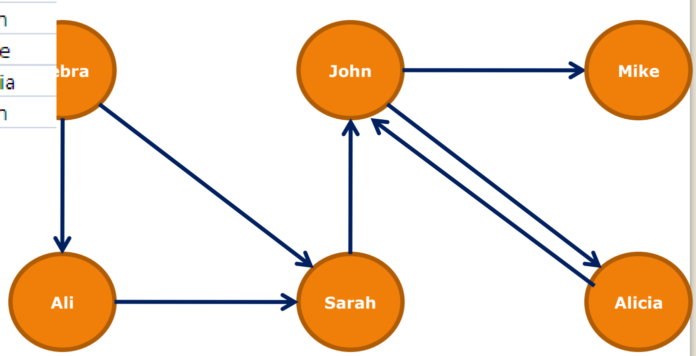
The WWW Is A Graph (2)

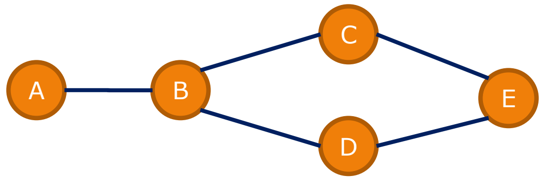
JT’s Extra: Applying Graphs To Web Design

- 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.

Graph Representation

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





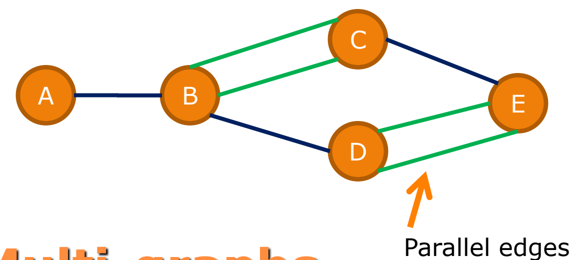
	A	B	C	D	E
A	0	1	0	0	0
B	1	0	1	1	0
C	0	1	0	0	1
D	0	1	0	0	1
E	0	0	1	1	0

Graph Representation

- Graphs Up To This Point:
  - Do not allow for multiple edges between any pair of vertices.
- Multi-graphs are a type of graph that allows for these cases.

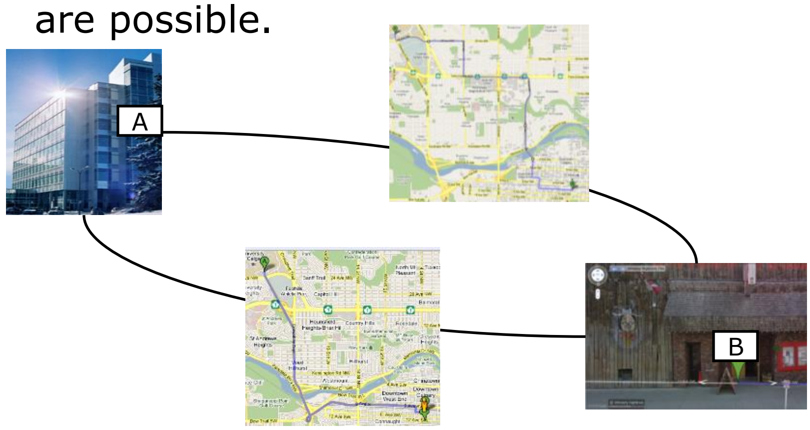
JT's Extra: Repeated Connections

- In a multi-graph the set of edges is a multi-set (JT: recall a multi-set is a set that allows for duplications)
- Edges can be repeated
- Graphs are special cases of multi-graphs

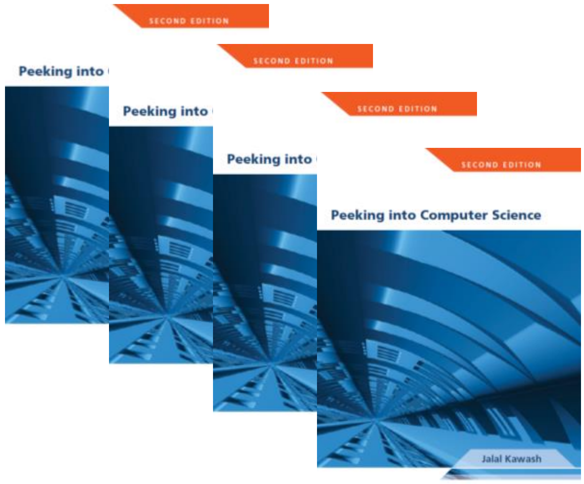


Multi-graphs

- Example: path finding when alternatives are possible.



JT's Extra: Application Multi-graphs



Euler Paths

34

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

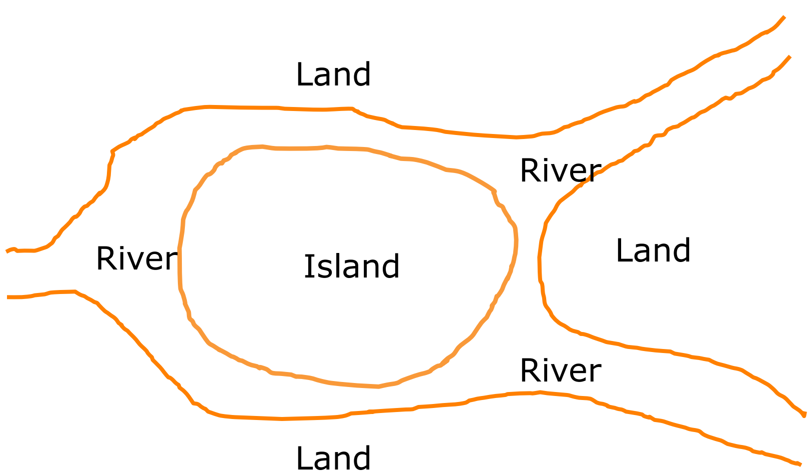
Objectives

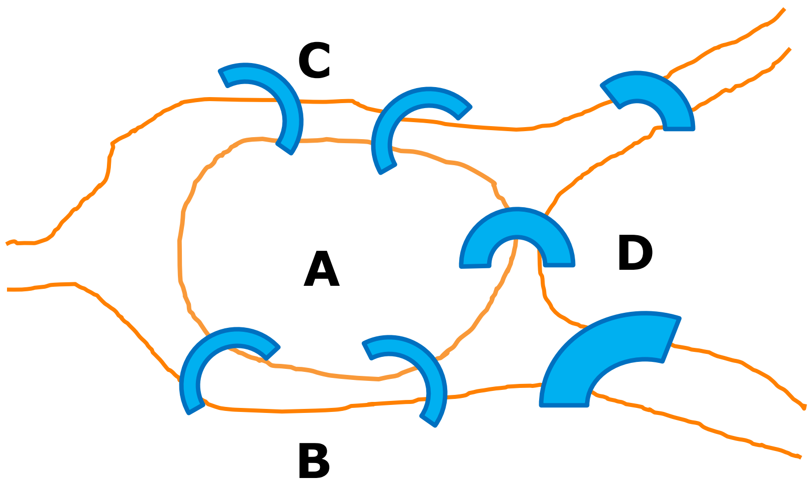
Peeking into Computer Science 35

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

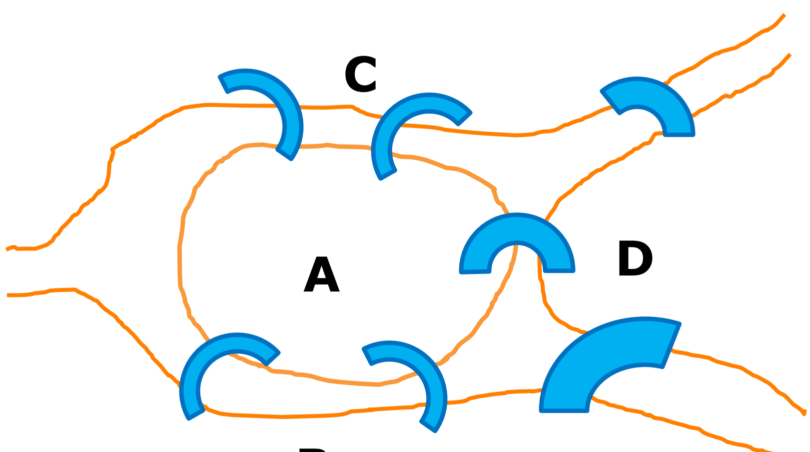
- They wrote to Swiss Mathematician Leonhard Euler for help

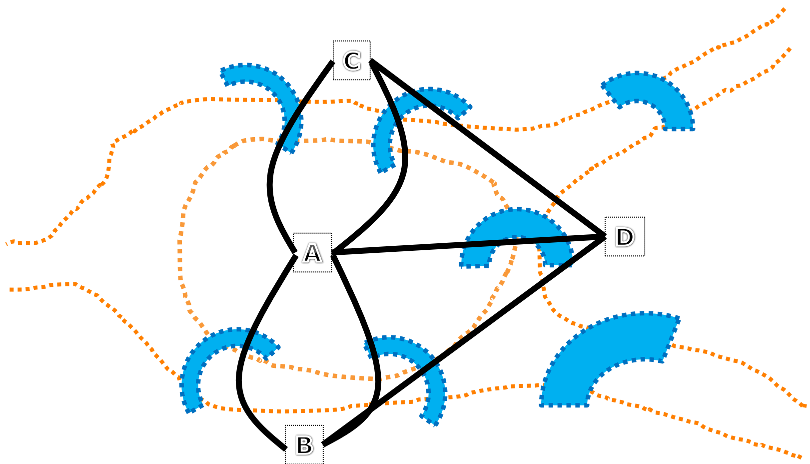
Konigsberg's Walk

Leonhard Euler

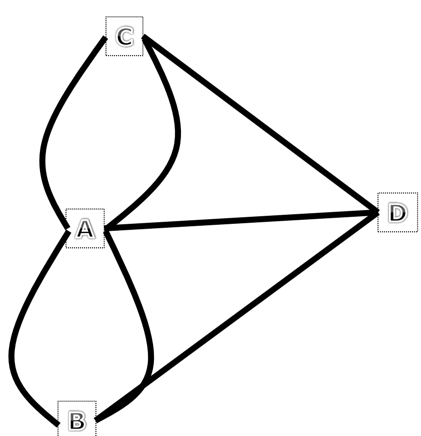
- 1707 –1783
- The greatest mathematician of the 18th century and one of the greatest of all time
- a pioneering Swiss mathematician and physicist
- important discoveries in graph theory
- introduced much of the modern mathematical terminology and notation

Konigsberg Multigraph

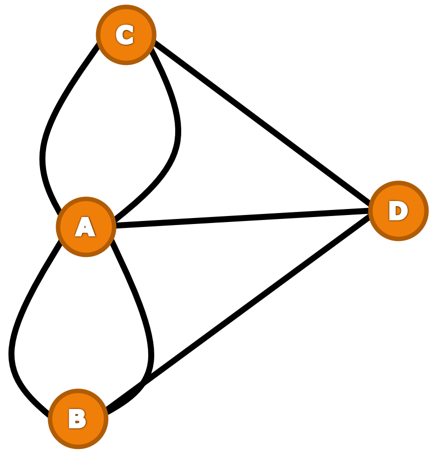




Konigsberg multigraph



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**

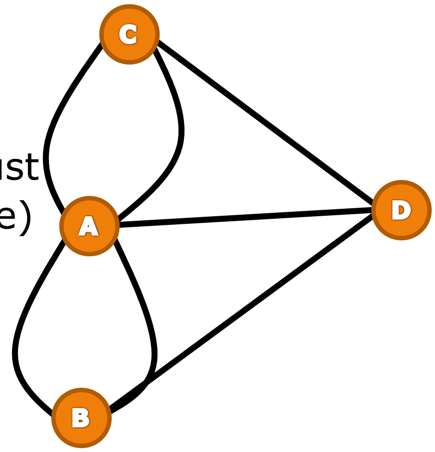
- Such a walk is impossible on any graph as long as the graph has one vertex with an *odd* degree

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)

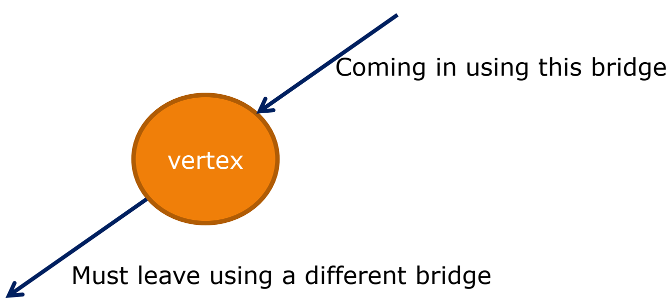
Success!



Peeking into Computer Science © Jalal Kawash 2010 46

Even-Degree Vertex (JT: Not Start/End)

- To cross every bridge once

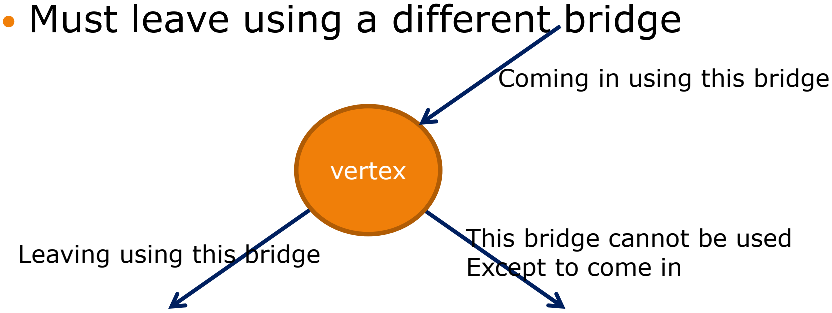


- Each vertex must have an even degree

Peeking into Computer Science © Jalal Kawash 2010 47

Odd-Degree Vertex (JT: Not Start/End)

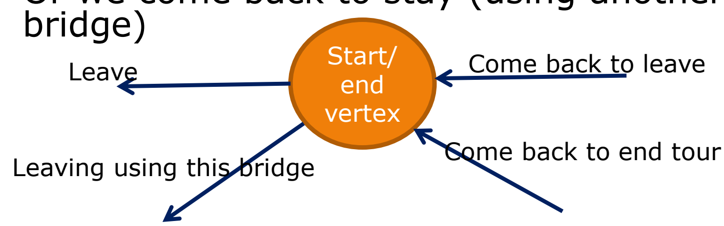
- Must come in using some bridge
- Must leave using a different bridge



- We're stuck

The Starting/Ending 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

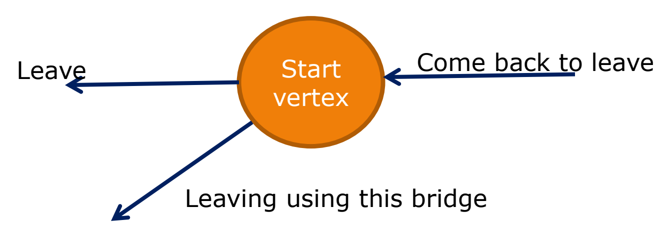
An Easier Problem

- 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

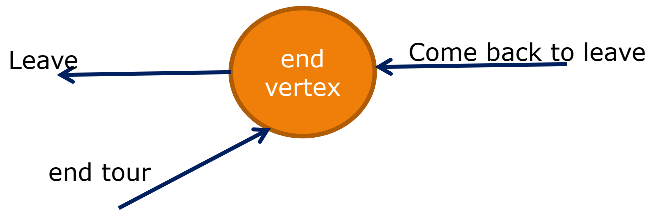
Euler Path – Start Vertex

- The starting vertex must have an odd degree



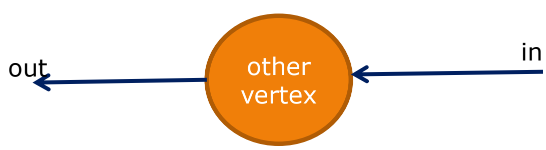
- **Never** come back to stay

- The end vertex must have an odd degree



Euler Path – End Vertex

- All other vertices must have an even degree

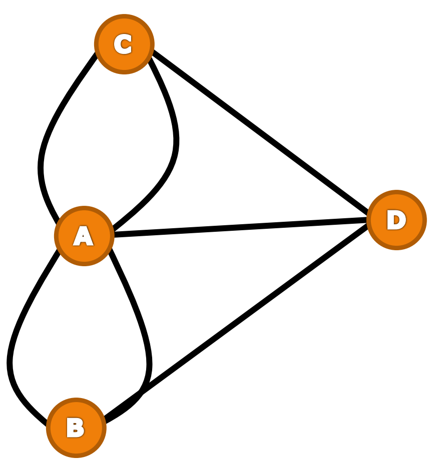


- Every **in** must have a matching **out** on "new" bridges

Other Vertices

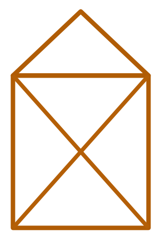
- Start and end vertices must have odd degrees
- Every other vertex must have an even degree
- There is no Euler Path for the Konigsberg graph

Euler (non-cycle) Path Requirements

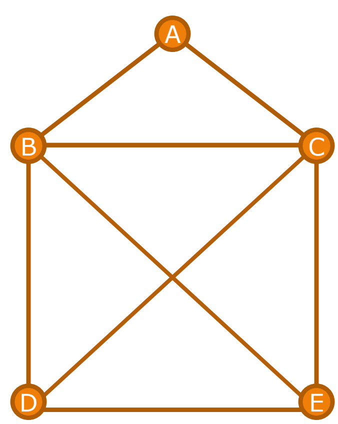


No Euler Path for this Graph

- 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

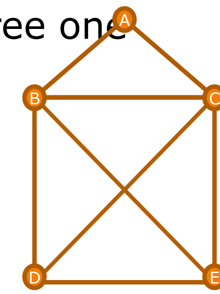


A Similar Problem



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

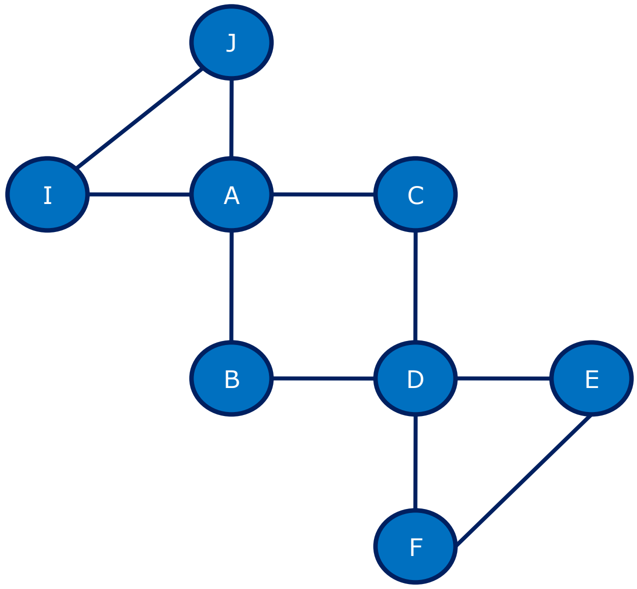


Graph Representation

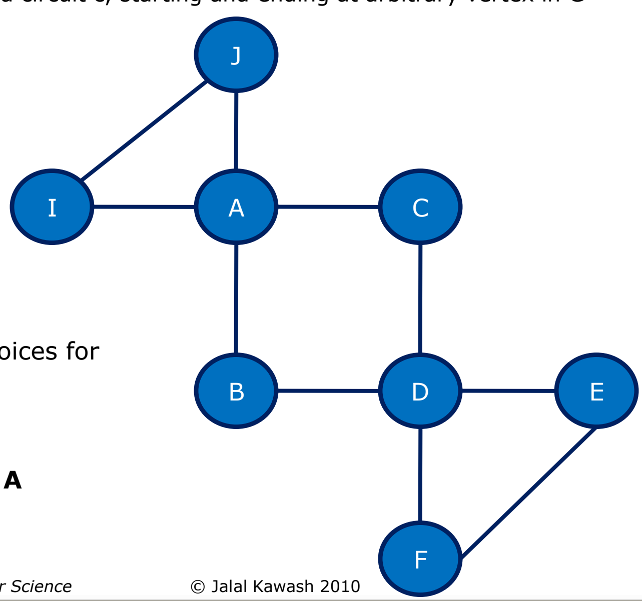
Given a graph G (all vertices have even degrees):

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



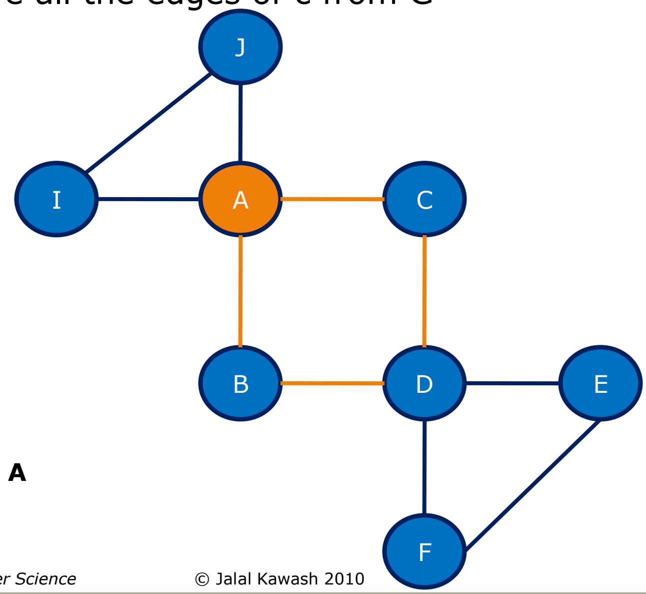
1. Construct a circuit c, starting and ending at arbitrary vertex in G



Note: other choices for c are possible

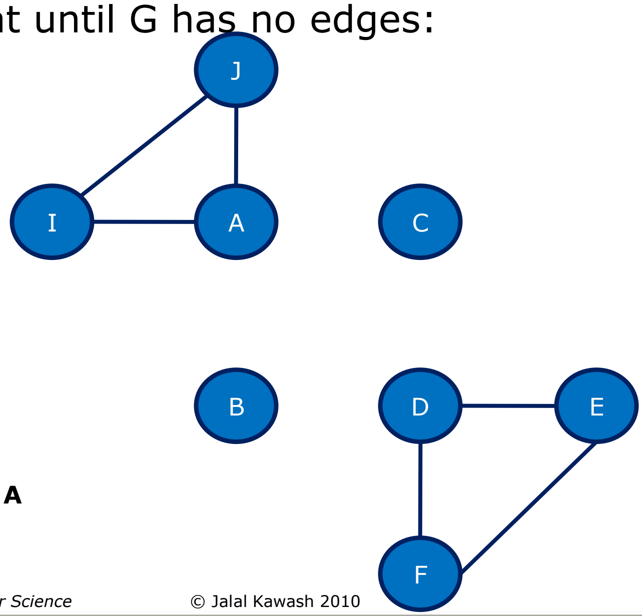
c: A, B, D, C, A

2. Remove all the edges of c from G



c: A, B, D, C, A

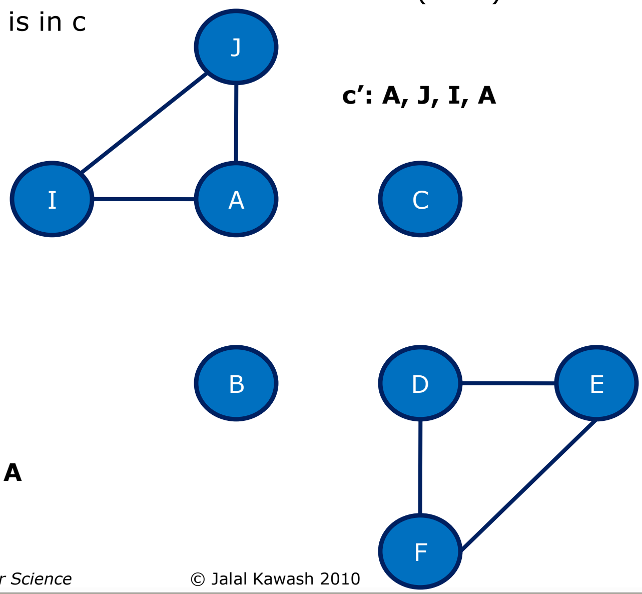
3. Repeat until G has no edges:



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

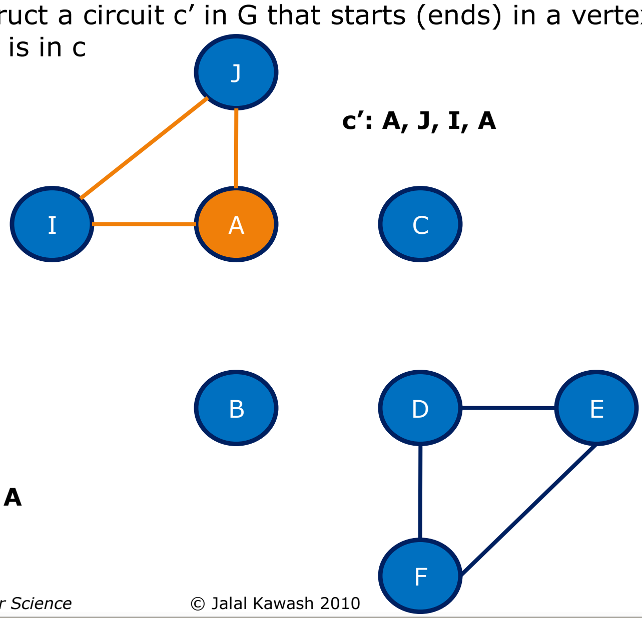


c: A, B, D, C, A

Peeking into Computer Science © Jalal Kawash 2010 64

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

c′: A, J, I, A

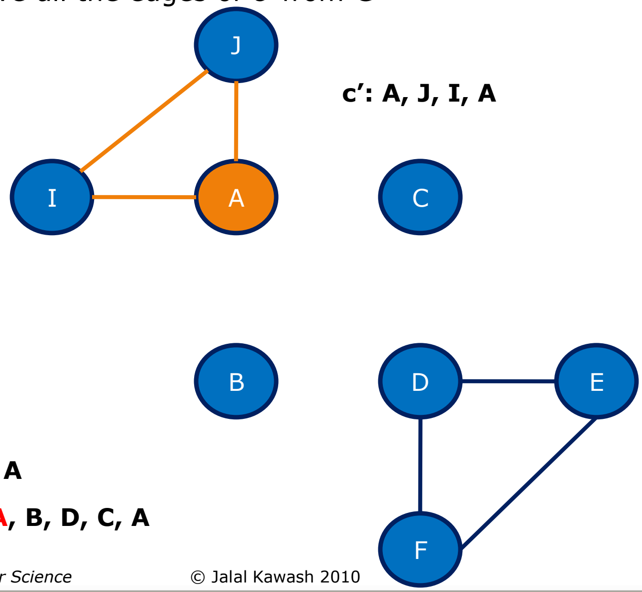


c: A, B, D, C, A

Peeking into Computer Science © Jalal Kawash 2010 65

a) Add c’ to c at v
b) Remove all the edges of c’ from G

c’: A, J, I, A

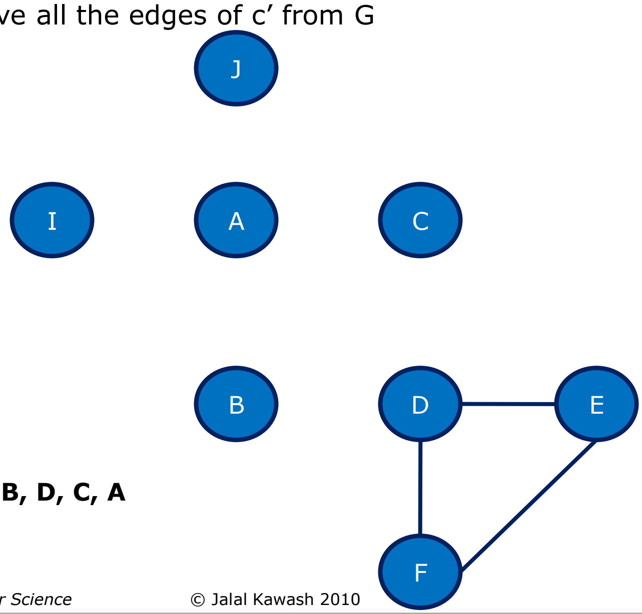


c: A, B, D, C, A

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

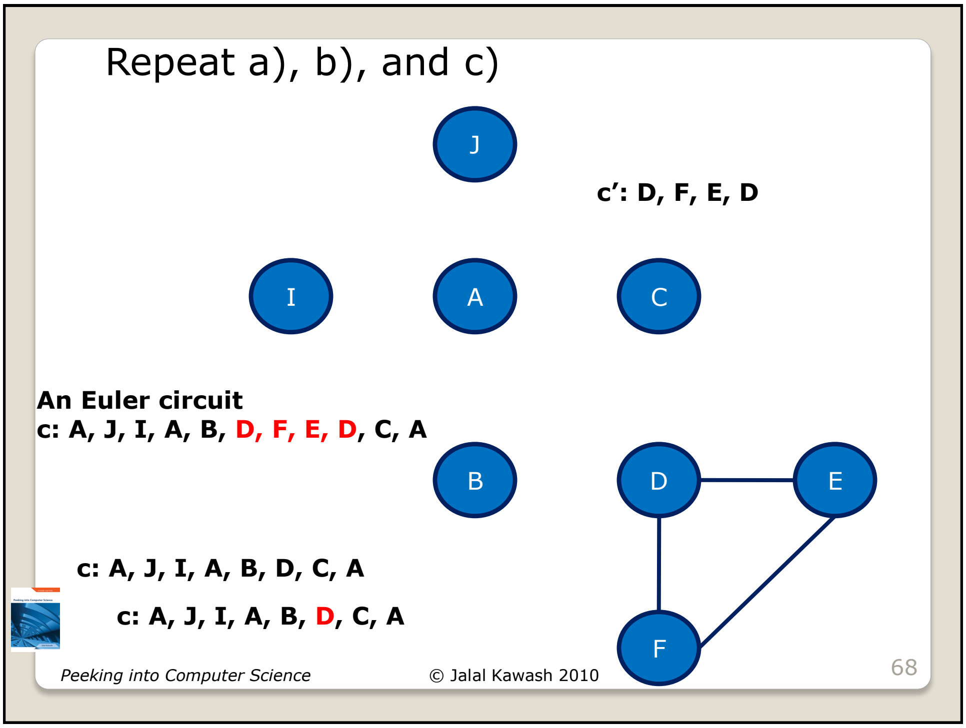
Peeking into Computer Science © Jalal Kawash 2010 66

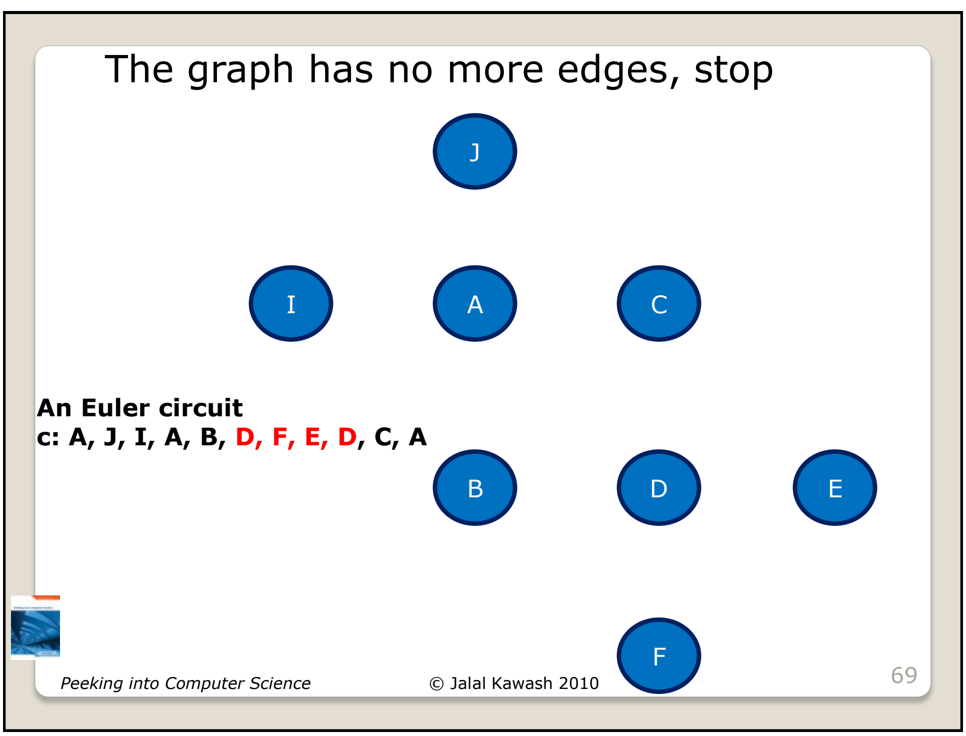
a) Add c’ to c at v
b) Remove all the edges of c’ from G

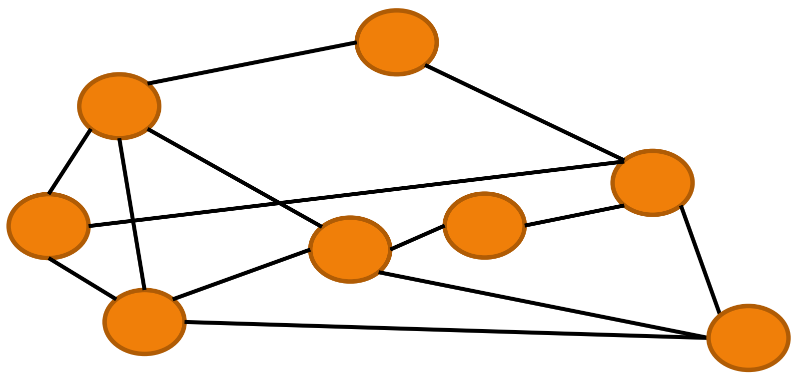


c: A, J, I, A, B, D, C, A

Peeking into Computer Science © Jalal Kawash 2010 67


Repeat a), b), and c)
J
c': D, F, E, D
I
A
C
An Euler circuit
c: A, J, I, A, B, D, F, E, D, C, A
B
D
E
c: A, J, I, A, B, D, C, A
c: A, J, I, A, B, D, C, A
F
Peeking into Computer Science
© Jalal Kawash 2010
68


The graph has no more edges, stop
J
I
A
C
An Euler circuit
c: A, J, I, A, B, D, F, E, D, C, A
B
D
E
F
Peeking into Computer Science
© Jalal Kawash 2010
69



JT's Extra: Euler Cycle Or Path?