CS1113
Circuits in Graphs

Lecturer:

Professor Barry O'Sullivan

Office: 2.65, Western Gateway Building

email: *b.osullivan@cs.ucc.ie*

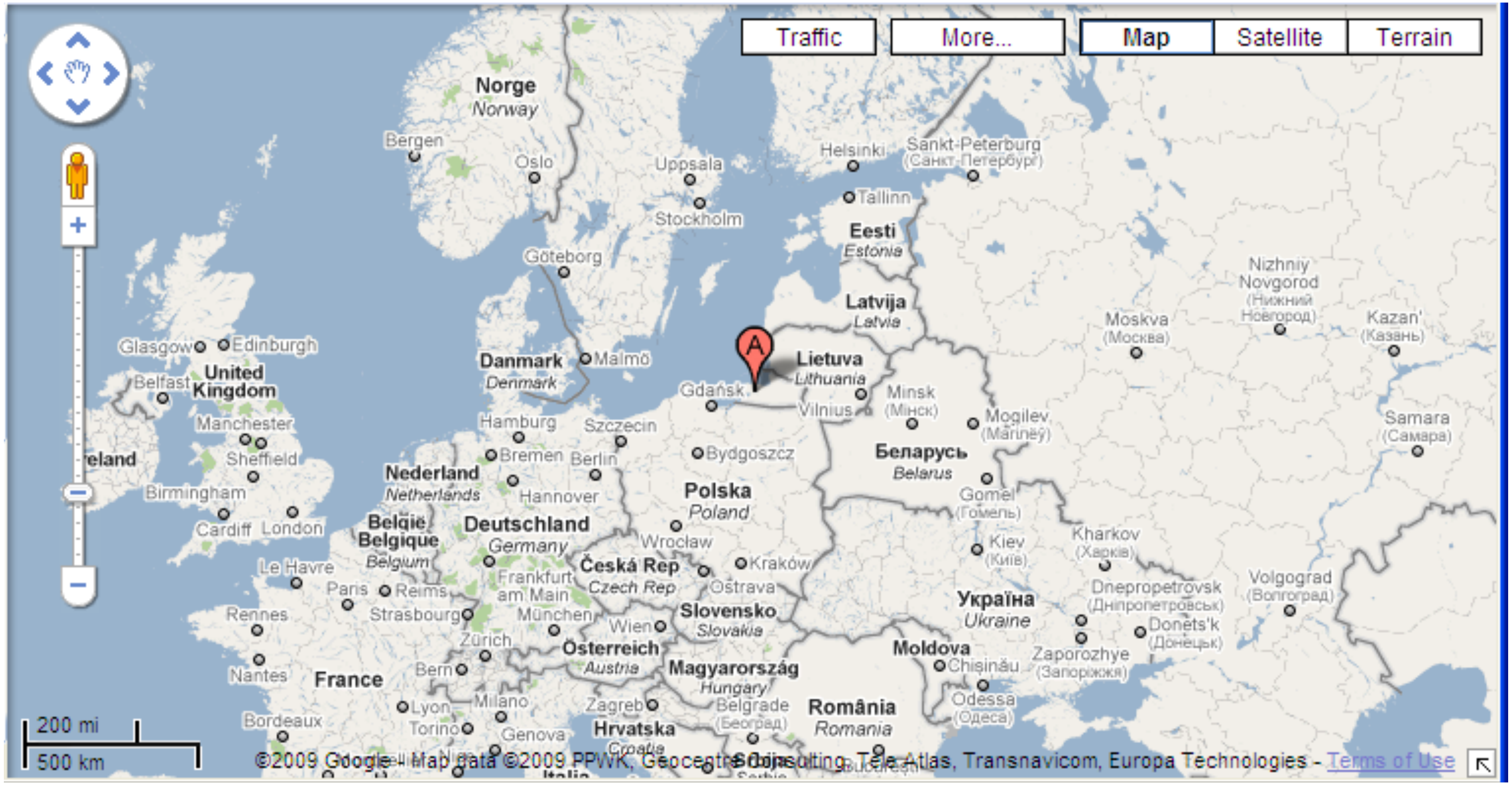
http://osullivan.ucc.ie/teaching/cs1113/

Circuits

A walk round the bridges of Königsberg

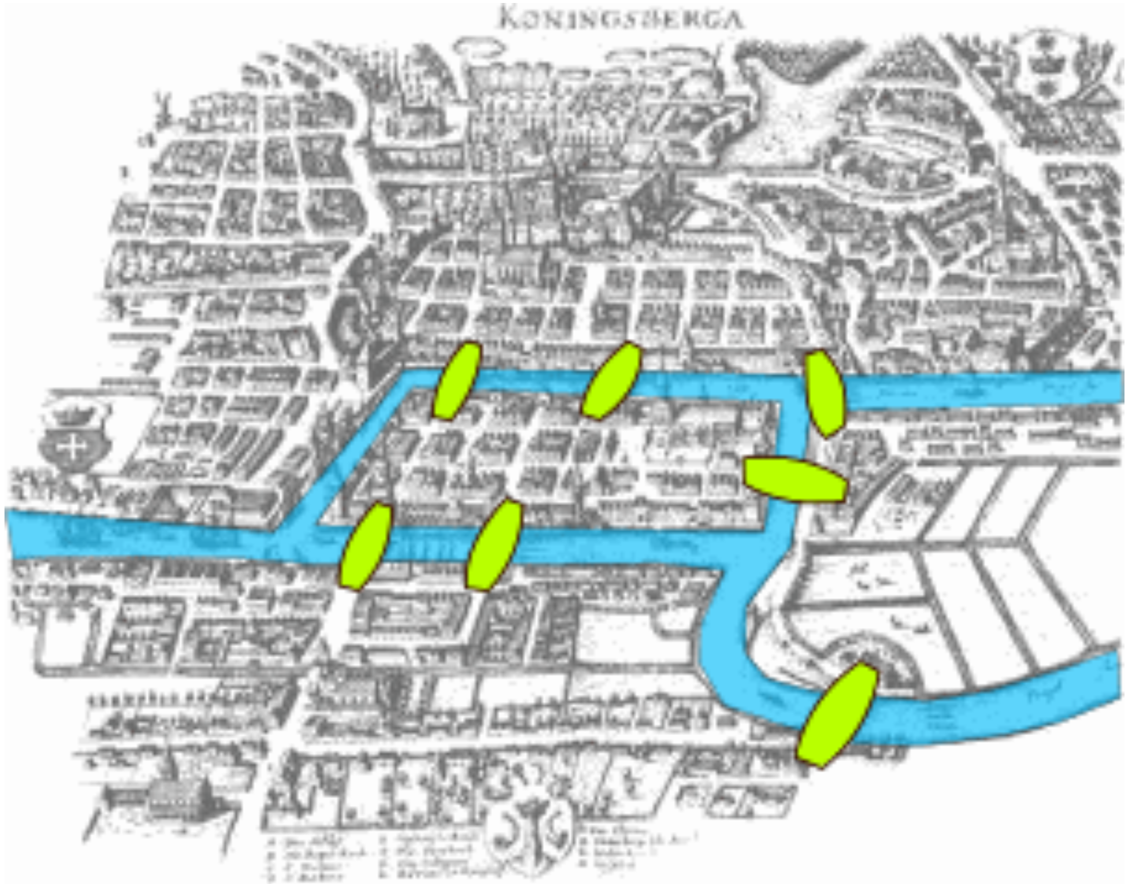
...and...

circuits
Euler circuits
Hamiltonian circuits
representing graphs
subgraphs
connected graphs


Traffic
More...
Map
Satellite
Terrain
Norge
Norway
Bergen
Oslo
Uppsala
Stockholm
Helsinki
Sankt-Peterburg
Tallinn
Eesti
Estonia
Göteborg
Latvija
Latvia
Moskva
Nizhniy Novgorod
Kazan'
Danmark
Denmark
Malmö
Lietuva
Lithuania
Gdańsk
Vilnius
Minsk
Mogilev
Glasgow
Edinburgh
Belfast
United Kingdom
Manchester
Sheffield
Birmingham
Cardiff
London
Hamburg
Szczecin
Bremen
Berlin
Bydgoszcz
Беларусь
Belarus
Samara
Nederland
Netherlands
Hannover
Polska
Poland
Gomel
België
Belgique
Belgium
Deutschland
Germany
Wrocław
Kraków
Kharkov
Kiev
Le Havre
Paris
Reims
Frankfurt am Main
Česká Rep
Czech Rep
Ostrava
Україна
Ukraine
Dnepropetrovsk
Volgograd
Rennes
Strasbourg
München
Wien
Slovensko
Slovakia
Donets'k
Zürich
Österreich
Austria
Moldova
Chişinău
Zaporozhye
Nantes
France
Bern
Magyarország
Hungary
Odessa
Milano
Lyon
Zagreb
Belgrade
România
Romania
Bordeaux
Torino
Genova
Hrvatska
200 mi
500 km
©2009 Google - Map data ©2009 PPWK, Geocentre Consulting, Tele Atlas, Transnavicom, Europa Technologies - Terms of Use

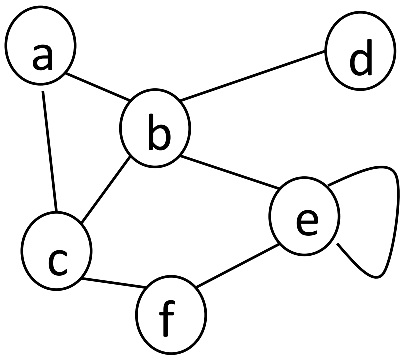
The Bridges of Königsberg

The residents of Königsberg liked to take a walk on Sunday afternoons. Was it possible for them to cross every bridge exactly once, and return to their starting point?



Circuits

A circuit is a path that starts and ends at the same vertex.

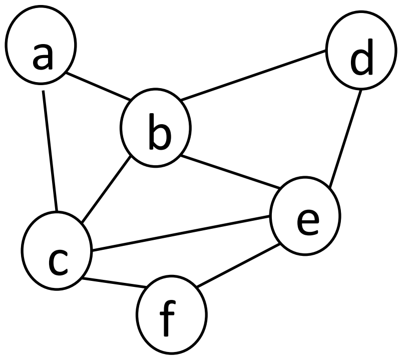


<(a,b),(b,c),(c,a)>
<(f,e),(e,e),(e,f),(f,c),(c,b),(b,a),(a,c),(c,f)>
<(a,b),(b,a)>
<(e,e)>
are all circuits

<(d,b),(f,c)>
<(d,b),(b,c),(c,f)>
<(c,b),(b,a),(b,c),(c,b))>
are not circuits

Euler Circuits

An Euler circuit is a circuit that contains every edge in the graph exactly once.



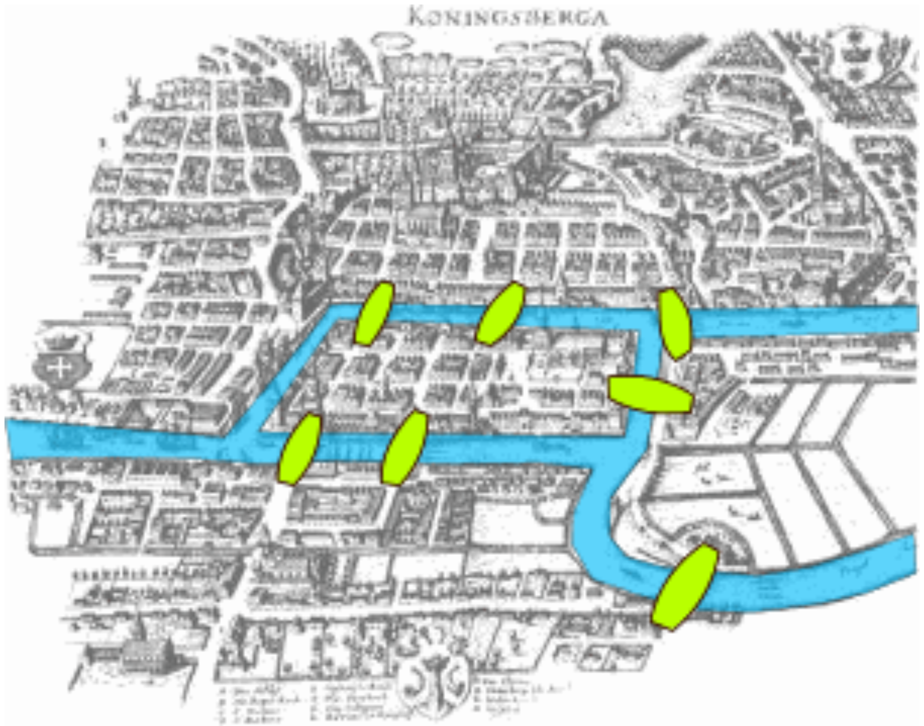
<(c,b),(b,a),(a,c),(c,f),(f,e),(e,d),(d,b),(b,e),(e,c)>
<(b,e),(e,d),(d,b),(b,c),(c,e),(e,f),(f,c),(c,a),(a,b)>
are both Euler circuits

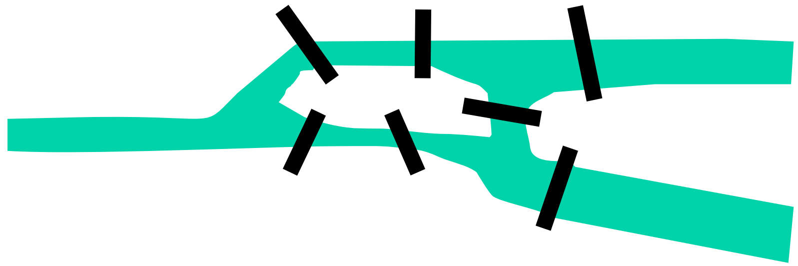
<(a,c),(c,f),(f,e),(e,d),(d,b),(b,a)>
is not an Euler circuit

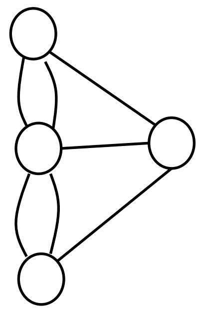
An Euler path is a path that contains every edge in the graph exactly once.

The Bridges of Königsberg

The residents of Königsberg like to take a walk on Sunday afternoons. Is it possible for them to cross every bridge exactly once, and return to their starting point?

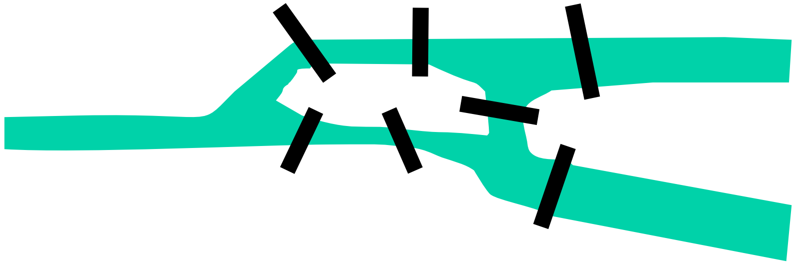


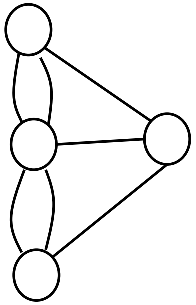




Is there an Euler circuit for the above multigraph?

Euler Circuits

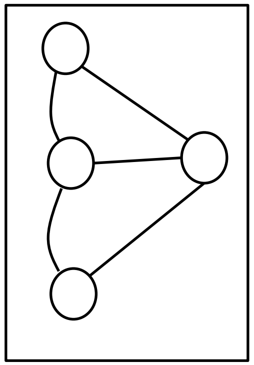




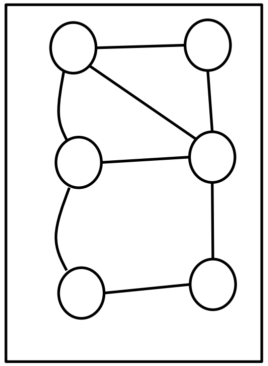
An Euler circuit visits every edge exactly once, and starts and finishes at the same vertex

Is there an Euler circuit for the above multigraph? No

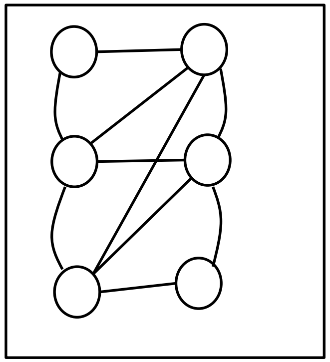
How about these?



No



No

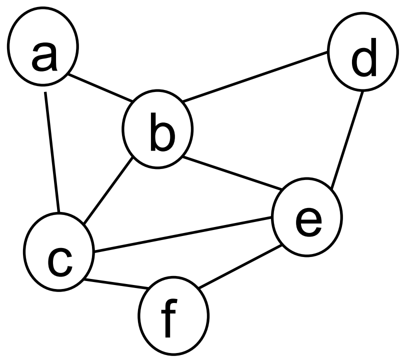


yes

A graph has an Euler circuit if and only if every vertex in the graph has even degree.

Hamiltonian Circuits

A Hamiltonian circuit is a circuit that contains every vertex in the graph exactly once.



<*(a,c),(c,f),(f,e),(e,d),(d,b),(b,a)*>
is a Hamiltonian circuit

<(*c,b*),(*b,a*),(*a,c*),(*c,f*),(*f,e*),(*e,d*),(*d,b*),(*b,e*),(*e,c*)>
is not a Hamiltonian circuit (it visits *c*, *b* and *e* twice)

<*(d,e),(e,f),(f,c),(c,b),(b,a)*>
is not a Hamiltonian circuit (it starts and finishes at different vertices)

A Hamiltonian path is a path that contains every vertex in the graph exactly once.

William Hamilton

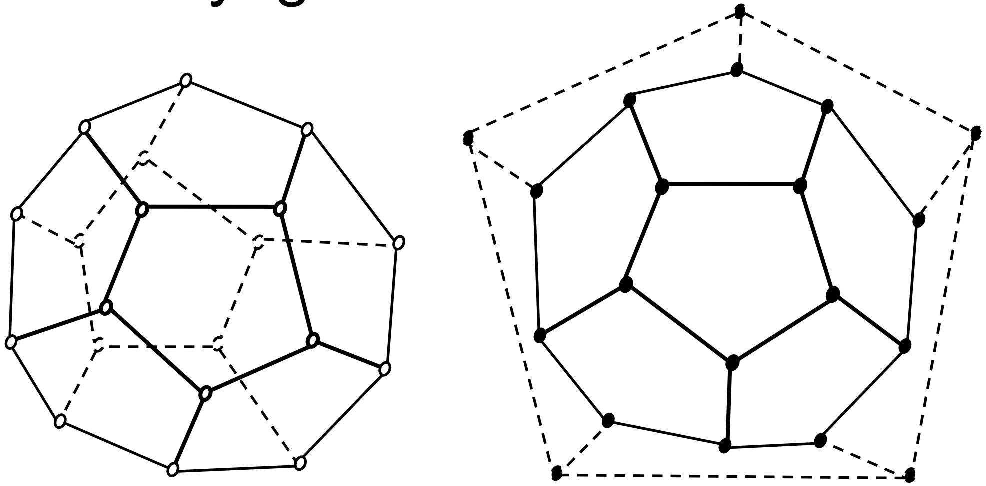
William Rowan Hamilton

Born	August 4, 1805 Dublin, Ireland
Died	September 2, 1865 (aged 60) Dublin, Ireland
Residence	Ireland
Nationality	Irish
Field	Mathematician, physicist, and astronomer
Institutions	Trinity College Dublin
Alma mater	Trinity College Dublin

from wikipedia

There is no simple method for deciding whether or not a graph has a Hamiltonian circuit – we have to search for one.

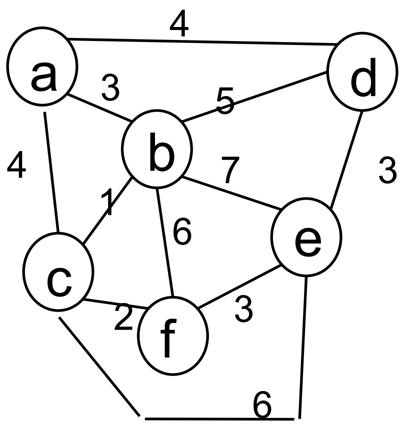
"A voyage around the world"

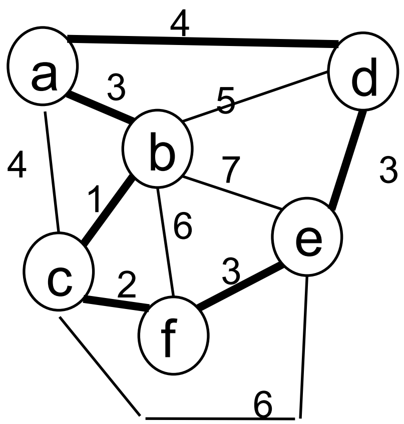


Exercise: find a circuit that visits every vertex exactly once.

Hamiltonian Circuits and TSP

The idea of the Hamiltonian circuit is the basis of the travelling salesman problem





There is no known efficient algorithm for solving the TSP

Find the cheapest circuit that visits every city, but doesn't visit any city twice.

We will see the travelling salesman problem in later lectures.

Representing Graphs

If graphs are ubiquitous in computer science and applications, then we need a way of representing them in programs

We defined graphs in terms of *sets* of vertices, and *sets* of edges, where the edges might themselves be *sets*: this is all we needed for understanding their properties and algorithms

But for programming, we need a representation that makes it easy to find out which edges connect which vertices

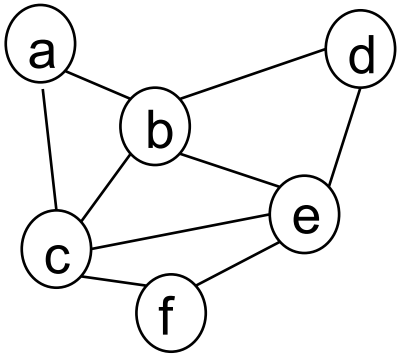
– we want it to allow fast lookup and response
– we want also want to minimise the amount of memory

There are two main methods:

– adjacency lists
– 2-dimensional arrays

Adjacency list representation

With each vertex, we associate a list containing all other vertices to which it is linked



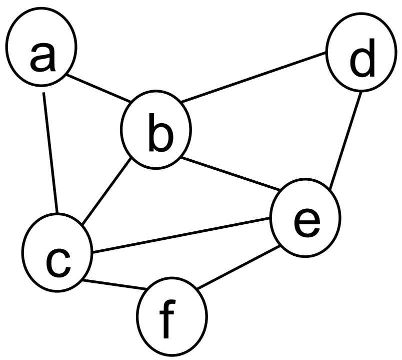
a: <b,c>
b: <a,c,d,e>
c: <a,b,e,f>
d: <b,e>
e: <b,c,d,f>
f: <c,e>

- easy to use for directed graphs
- easy to use for edge or vertex-weighted graphs
- can also be used for multigraphs
- only requires space for the edges that exist

but can be slow to process, since we have to search the adjacency list to find out whether or not two vertices are connected.

Table representation

- two dimensional array (or matrix, or table)
- the rows and columns are indexed by the vertices (so we have an *nxn* table for a graph with *n* vertices)
- each cell contains a "T" or a "1" if there is an edge between the corresponding vertices



	a	**b**	**c**	**d**	**e**	**f**
a		1	1			
b	1		1	1	1	
c	1	1			1	1
d		1			1	
e		1	1	1		1
f			1		1	

Properties of table representation

- in a simple graph, the table is symmetric
- in a directed graph, the table might not be symmetric

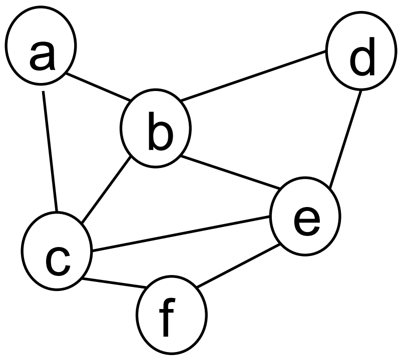
- for graphs with weights on the edges, we can represent the weight in the cell instead of "T" or "1"
- not good for representing multigraphs

- if the graph is sparse (i.e. not many edges), this wastes space

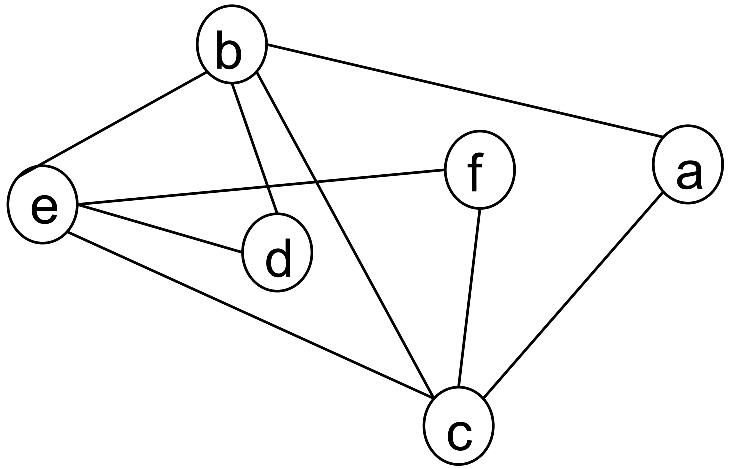
- but it is efficient to search – looking to see if there is an edge between two vertices *v* and *w* just involves looking at the cell [v][w].

The graph is not the sketch

- the way we draw the graph on the page does not change the underlying graph



and

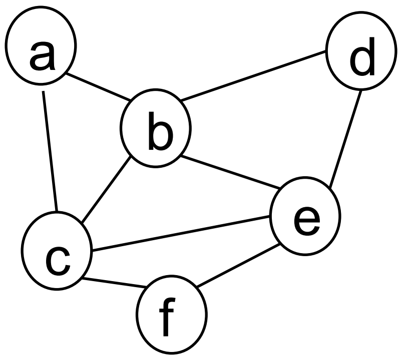


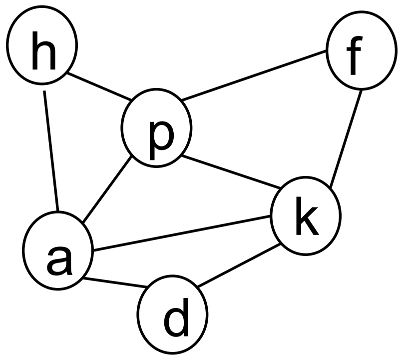
are the same graph. They will have an identical table representation.

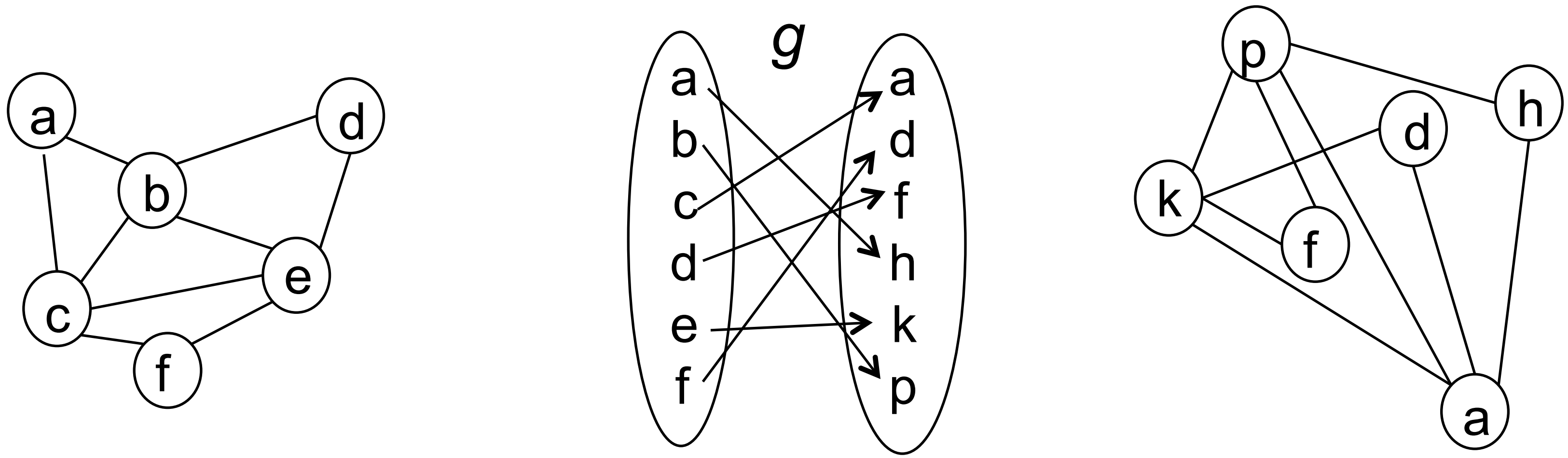
Isomorphic graphs

Suppose we have two graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$

Then G_1 and G_2 are isomorphic if and only if there is a bijective function g: $V_1 \rightarrow V_2$ such that there is an edge in E_1 between v_i and v_j if and only if there is an edge in E_2 between $g(v_i)$ and $g(v_j)$
i.e. if we can rename the vertices in G_1 so that G_1 and G_2 become the same graph

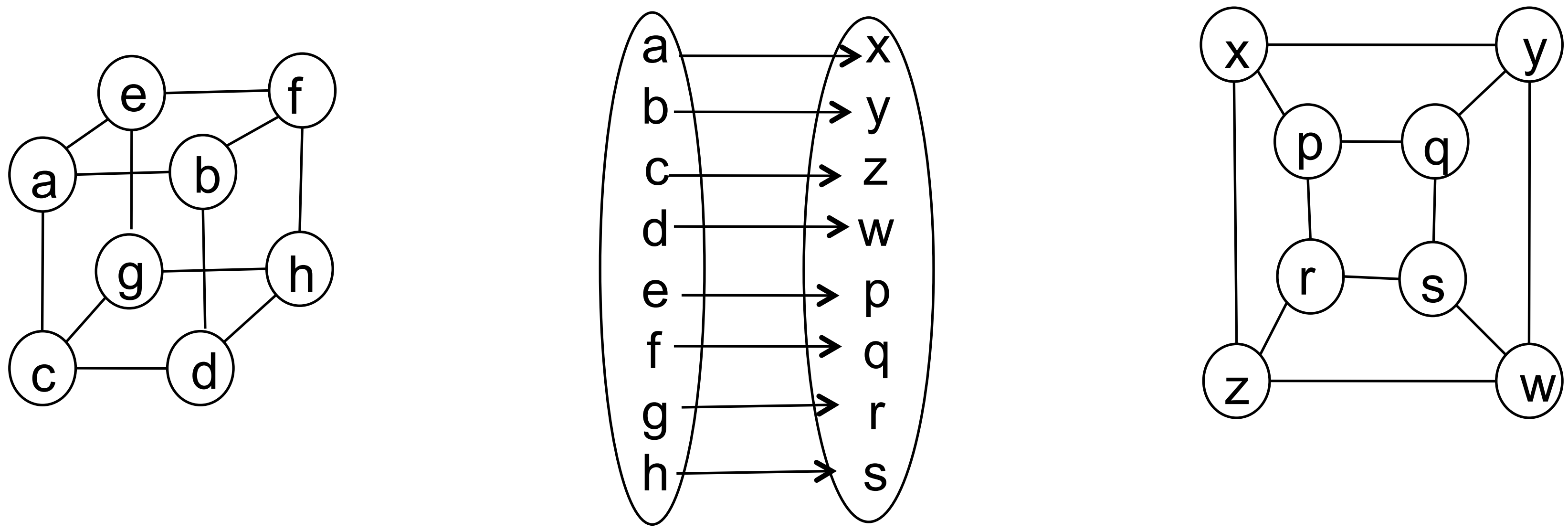






	a	**b**	**c**	**d**	**e**	**f**
a		1	1			
b	1		1	1	1	
c	1	1			1	1
d		1			1	
e		1	1	1		1
f			1		1	

	h	**p**	**a**	**f**	**k**	**d**
h		1	1			
p	1		1	1	1	
a	1	1			1	1
f		1			1	
k		1	1	1		1
d			1		1	

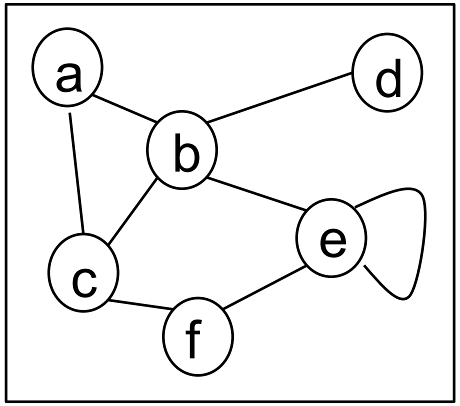


	a	b	c	d	e	f	g	h
a		1	1		1			
b	1			1		1		
c	1			1			1	
d		1	1					1
e	1					1	1	
f		1			1			1
g			1		1			1
h				1		1	1	

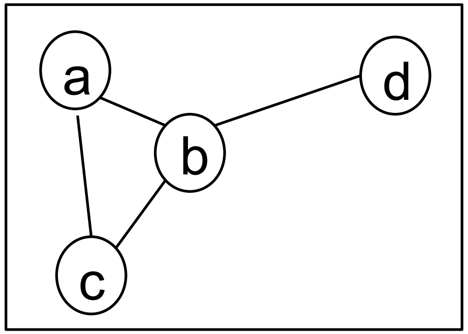
	x	y	z	w	p	q	r	s
x		1	1		1			
y	1			1		1		
z	1			1			1	
w		1	1					1
p	1					1	1	
q		1			1			1
r			1		1			1
s				1		1	1	

Subgraphs

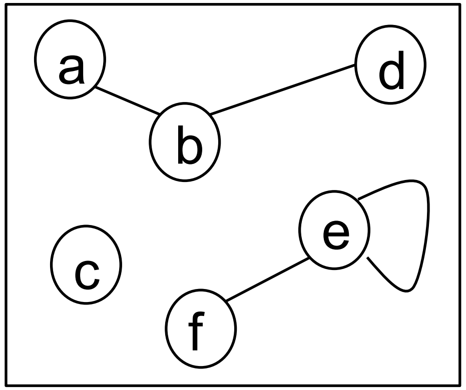
A subgraph H of a graph G=(V,E) is a graph H=(W,F), such that W ⊆ V, and F⊆E. Note that H is a graph, so every edge in F links vertices in W.



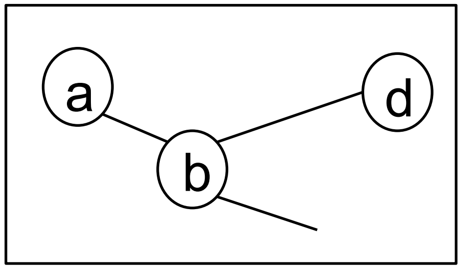
G



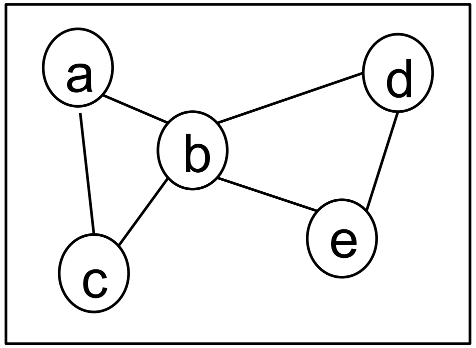
subgraph



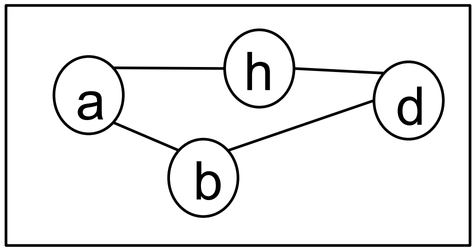
subgraph



NOT a subgraph



NOT a subgraph

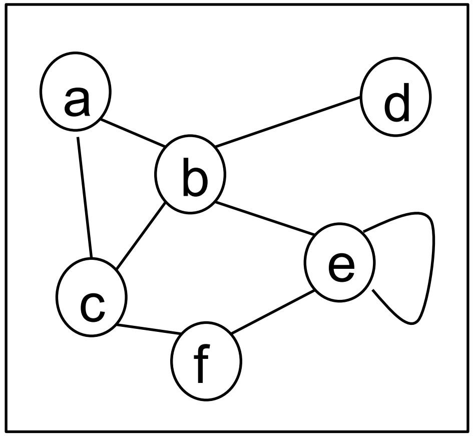


NOT a subgraph

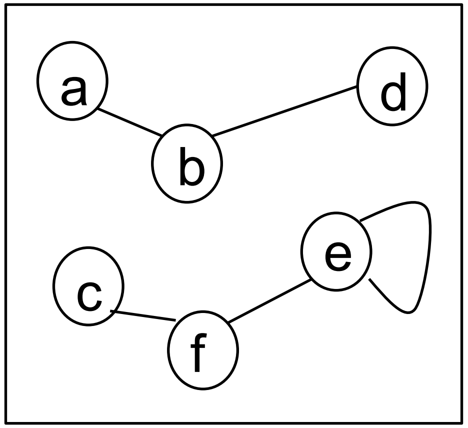
Connected graphs

reminder

A graph is connected if every pair of vertices *v*,*w* can be connected by a path which starts at *v* and ends at *w*.



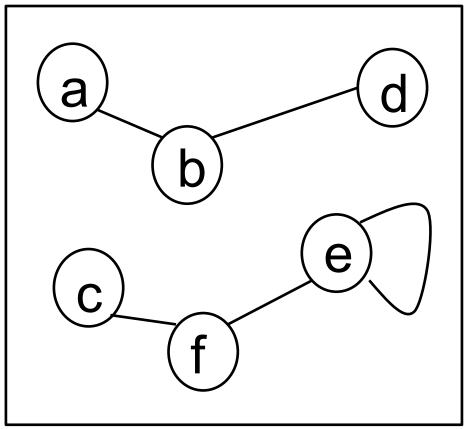
is a connected graph



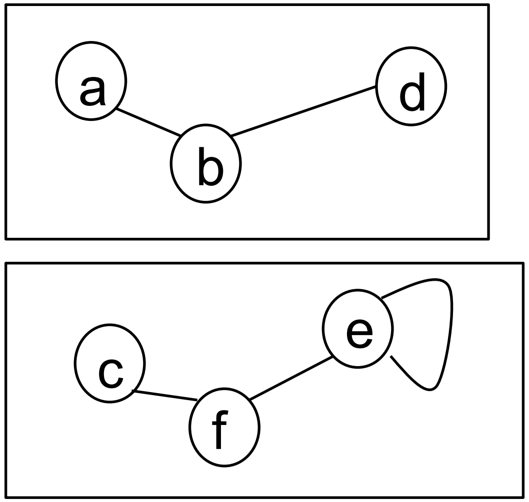
is not a connected graph – there is no path between, for example, *a* and *f*.

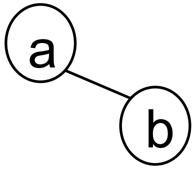
Connected graphs

A connected component, H, of a graph G is a connected subgraph of G that is not a proper subgraph of any other connected subgraph of G.
(we could say H is a maximal connected subgraph of G)



has two connected components:





is not a connected component (it is a connected subgraph, but not a maximal connected subgraph)

Next lecture ...

Trees

spanning trees

minimum spanning trees