Introduction to Engineering Systems, ESD.00

Networks

Lecture 7

Lecturers:
Professor Joseph Sussman
Dr. Afreen Siddiqi
TA: Regina Clewlow

Massachusetts Institute of Technology
Engineering Systems Division

The Bridges of Königsberg

- The town of Konigsberg in 18th century Prussia included two islands and seven bridges over the river Pregel.

- Residents had often thought about finding a walk such that starting from any of the four places, A,B,C,D, one crosses all of the seven bridges only once and then returns to the starting place.

- No one could find such a walk....

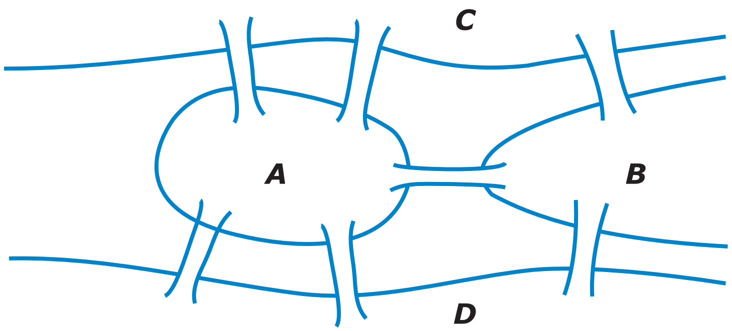


Image by MIT OpenCourseWare.

Figure: http://www.transtutors.com/homework-help/Discrete+Mathematics/Graph+Theory/konisberg-multigraph-bridge.aspx

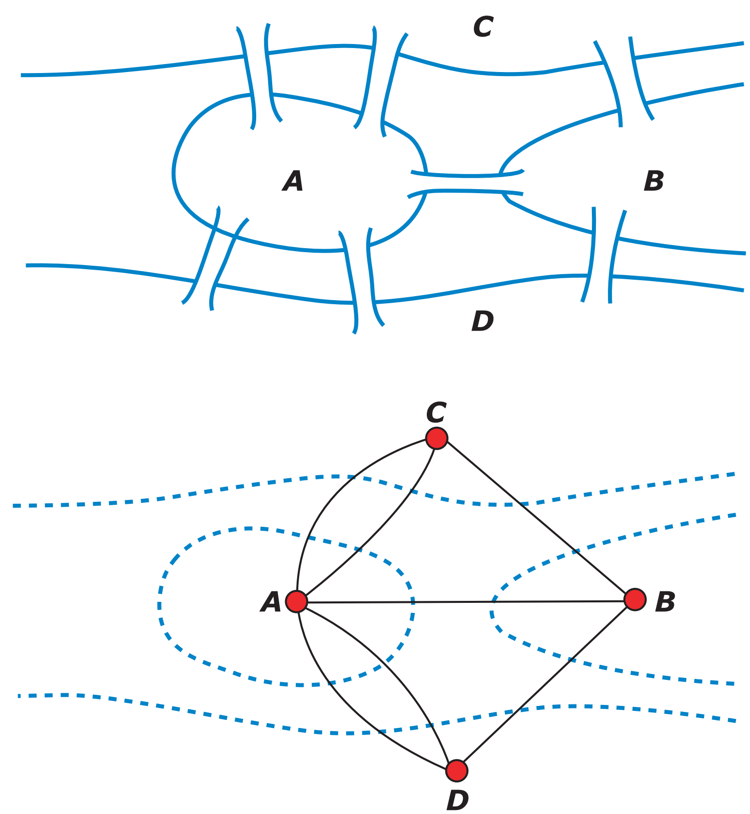
Graph Theory

- Leonhard Euler, in 1736, came up with the realization that this was not a problem of traditional geometry – measurements of angles, lengths, orientations do not matter.

- **The only two things that mattered were whether the islands or banks are connected by a bridge, and by how many bridges.**

- He modeled each place (island or bank) as a 'vertex' and each bridge as an 'edge' that connected the vertices.

- He mathematically proved that no such walk existed for the Koningsberg problem and founded an entirely new branch of mathematics along the way.



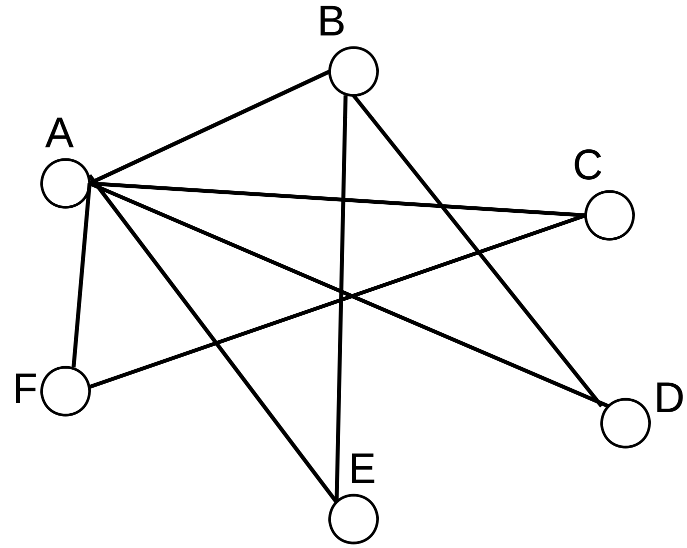
Images by MIT OpenCourseWare.

Figure: http://www.transtutors.com/homework-help/Discrete+Mathematics/Graph+Theory/konisberg-multigraph-bridge.aspx

Modeling Example - I

There were six people: A, B, C, D, E, and F in a party and following handshakes among them took place:

A shook hands with B, C, D, E and F
B, in addition, shook hands with D and E
C, in addition, shook hands with F

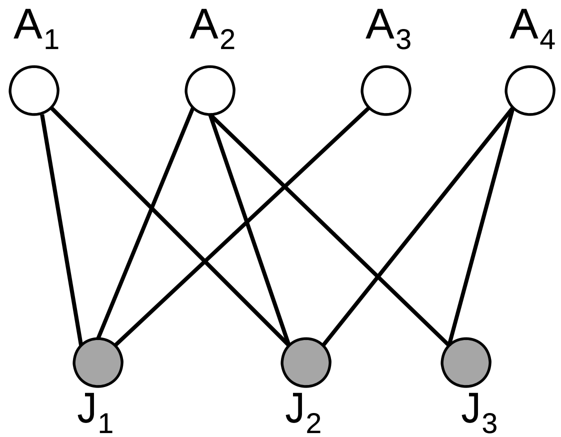


Ref [3]

Modeling Example - II

Consider a job application problem. There are three jobs J_1, J_2, J_3 for which four applicants A_1, A_2, A_3, A_4 have applied.

A_1 has applied for J_1 and J_2
A_2 has applied for J_1, J_2 and J_3
A_3 has applied for J_1
A_4 has applied for J_2 and J_3



Ref [3]

This type of graph is called a bi-partite graph.

A bi-partite graph has two types of vertices (nodes) and there are no edges between nodes of the same type.

Modeling Example - III

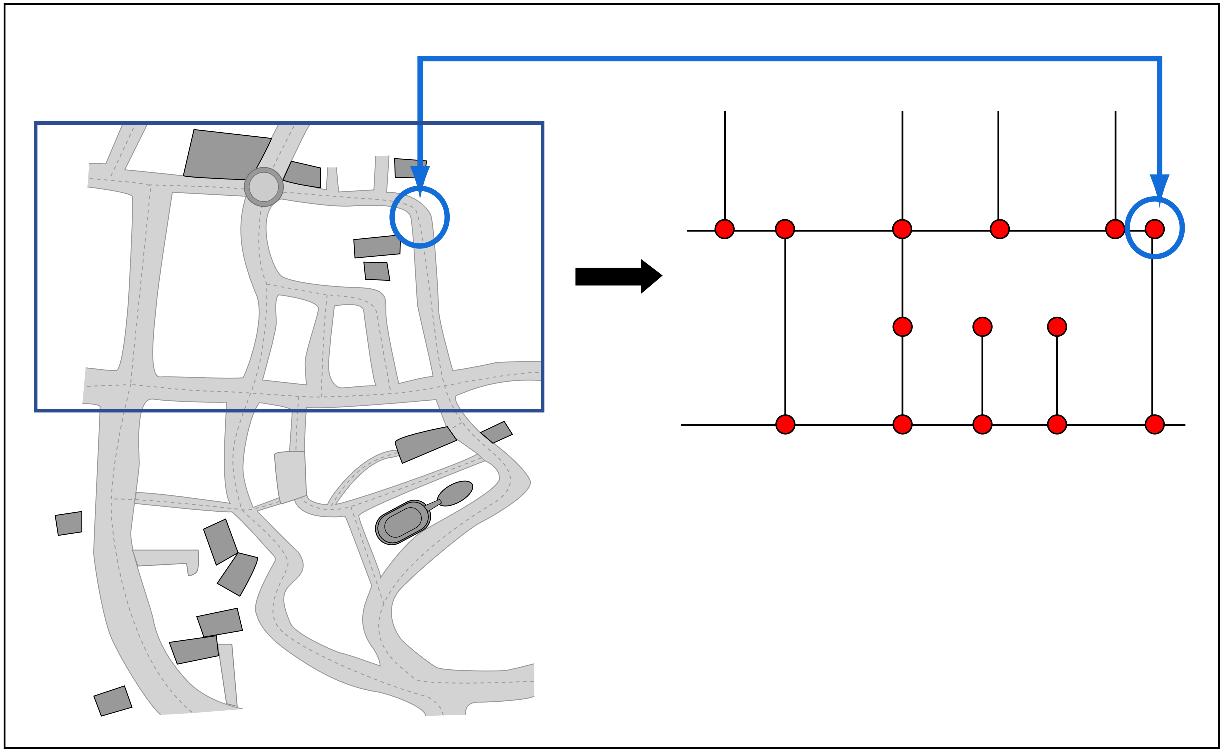


Image by MIT OpenCourseWare.

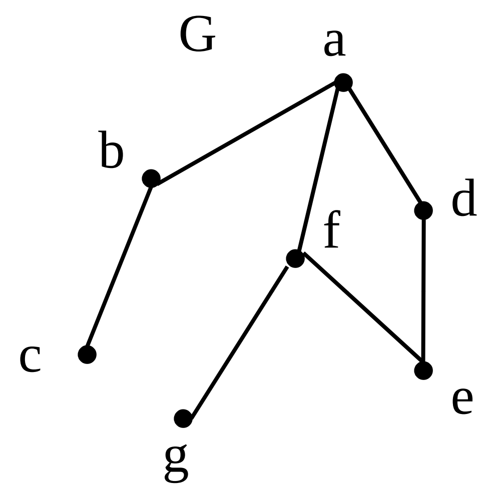
http://cvpr.uni-muenster.de/teaching/ws10/projektseminarWS10/

http://www.airlineroutemaps.com/USA/American_Airlines_caribbean.shtml

Graphs

Ref [3]

- A graph is a finite collection of vertices (or nodes) and edges (or links).
- To indicate a graph G has vertex set V and edge set E, we write G=(V,E)
- Each edge {x,y} of G is usually denoted by xy, or yx.
- What is the vertex set V(G) and edge set E(G) of the graph G shown?



V(G)={a, b, c, d, e, f, g}

E(G)={ab, bc, ad, de, af, fg, fe}

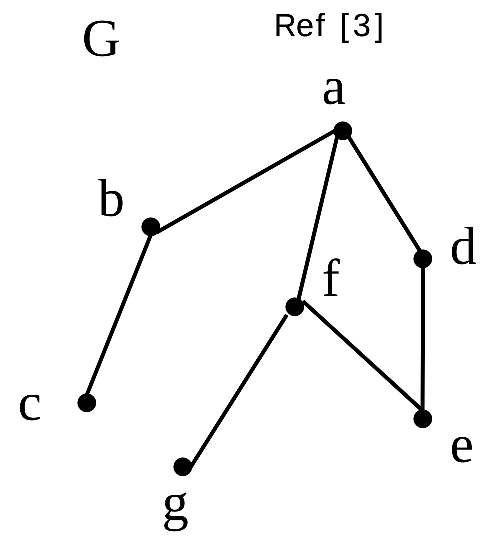
Adjacency

- If xy is an edge of G, the x and y are **adjacent vertices**

- Two adjacent vertices are referred to as **neighbors** of each other

- The set of neighbors of a vertex v is called the **open neighborhood** (or simply neighborhood) of v and is denoted as $N(v)$

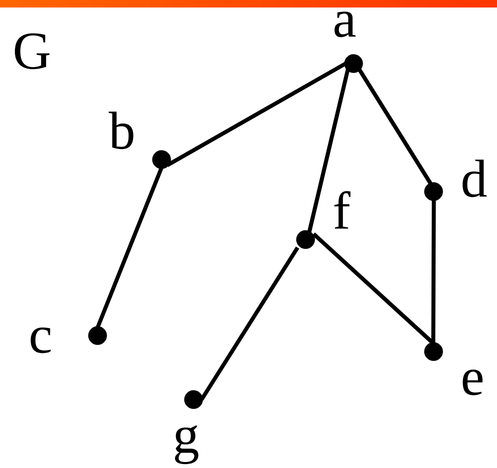
- For graph G shown on the right, what is:
- $N(a)$?
- $N(f)$?
- $N(b)$?



Order and Size

- The **number of vertices** in a graph G is the **order** of G
- The **number of edges** in G is the **size** of G
- The order of G (as shown on the right) is __
- The size of G (as shown on the right) is ___
- A graph of size 0 is called an empty graph – no two vertices are adjacent.
- A graph in which every pair of two vertices are adjacent is called a **complete graph**

Ref [3]



Multi-Graphs

- So far we've considered only zero or one edge between a pair of vertices
- What if there are more edges?
- Consider Euler's graph for the Köningsberg problem
- The graph K is a **multigraph**
- A multigraph has finite number of edges (including zero) between any two vertices
- So all graphs are multigraphs but not vice versa
- No loops are allowed in a multigraph – a vertex cannot connect to itself

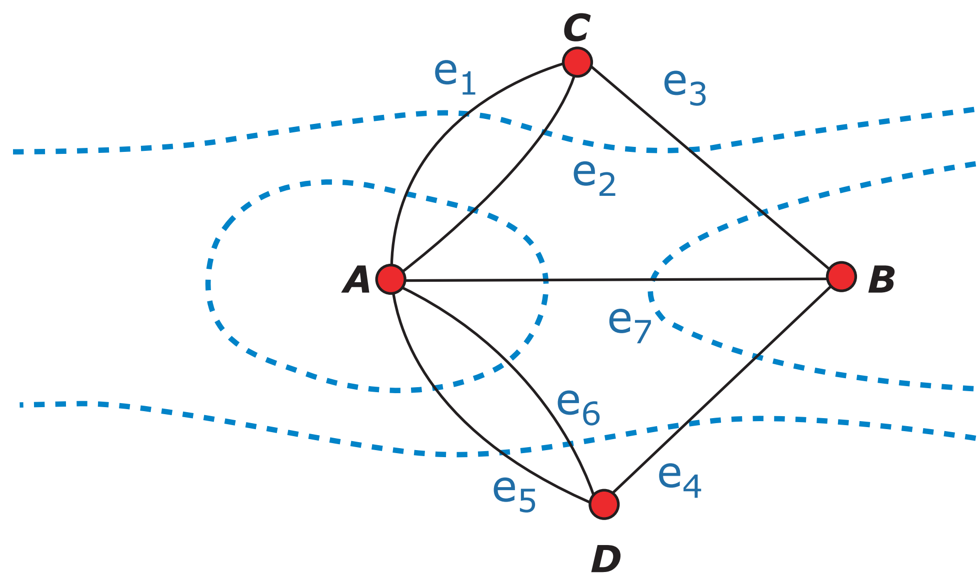


Image by MIT OpenCourseWare.

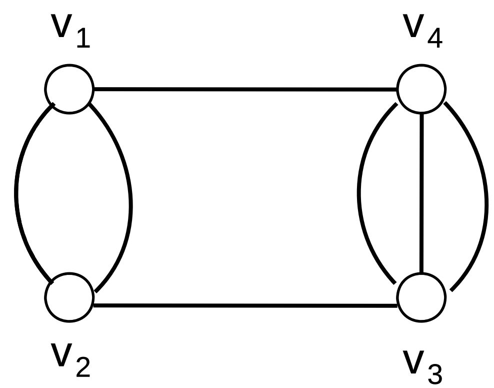
Adjacency Matrix

- In addition to set representation, we can use matrices to represent multigraphs

- We can create an **adjacency matrix** A such that its each (i,j) entry is the number of edges that exist between vertex i and vertex j

- For a multigraph G of order n with $V(G)=\{v_1, v_2, v_3..v_n\}$, the adjacency matrix of G is $n \times n$
  - $A(G) = [a_{ij}]_{nxn}$
  - where a_{ij}, the (i,j)-entry in $A(G)$ is the number of edges joining v_i and v_j

Ref [4]



$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 3 \\ 1 & 0 & 3 & 0 \end{bmatrix}$$

there are three edges between v_3 and v_4

Exercise

Draw G when G(A) is:

$$A = \begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 1 & 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 1 & 0 \end{bmatrix}$$

Why are the elements of the diagonal always zero?

What is the order (n) of G?

What is the size (m) of G?

How can you determine m from A?

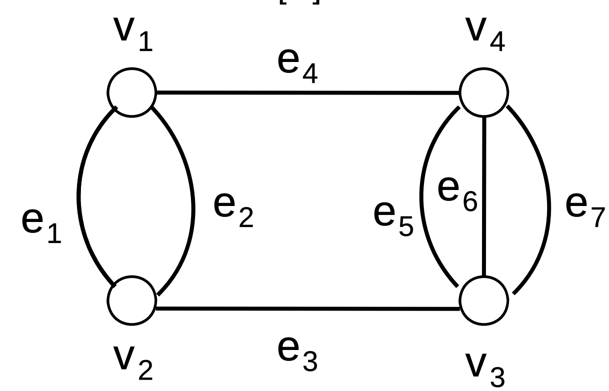
Incidence Matrix

- The Incidence Matrix, B is a binary, n x m matrix, where $b_{ij} = 1$ if v_i is an endvertex of edge e_j, otherwise it is zero.

- The incidence matrix contains more information than an adjacency matrix since it distinguishes between edges.

- Each column has two ones if each edge has two distinct vertices (i.e. when there are no loops and the graph is connected).

Ref [4]



$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

v_4 is an endvertex of e_4, e_5, e_6, and e_7

Vertex Degrees

- Given a vertex v in G, the degree of v in G, denoted by d_G (v), is defined as the number of edges incident with v.
- Which vertex has the highest degree in the Koningsberg problem? What is its degree?
- **In a multigraph, the sum of the degrees of its vertices is twice its size (number of edges).**

$$\boxed{\sum_{i=1}^{n} d(v_i) = 2m}$$

This is also known as the 'Hand-Shaking Theorem'

- A vertex with the highest degree is called a **hub** in a graph (or network).

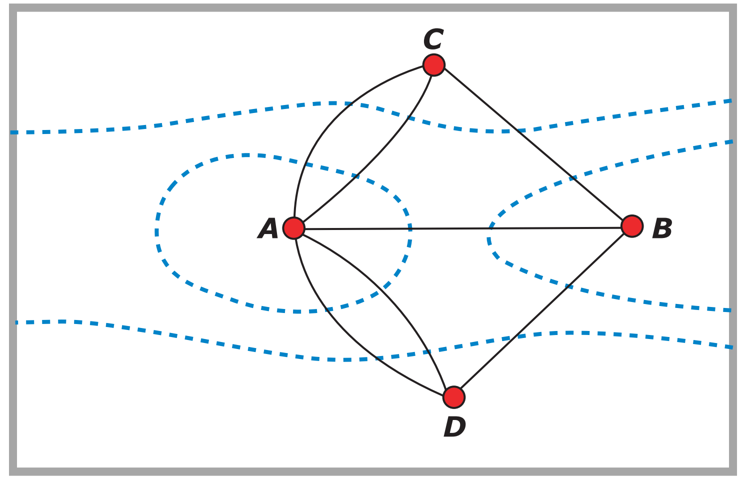
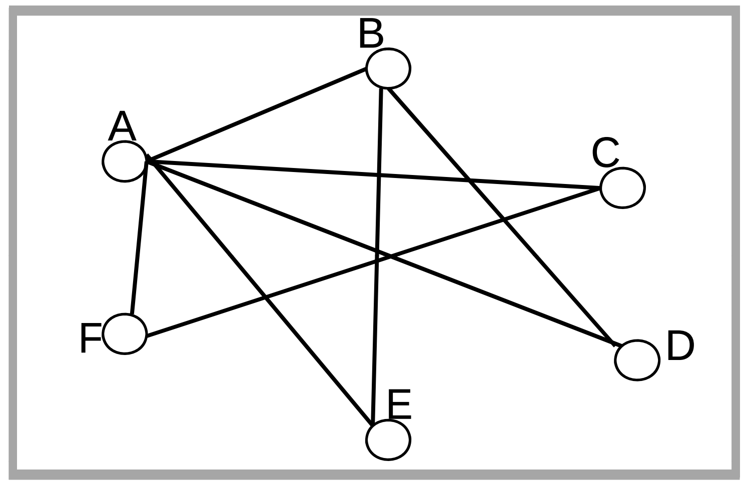


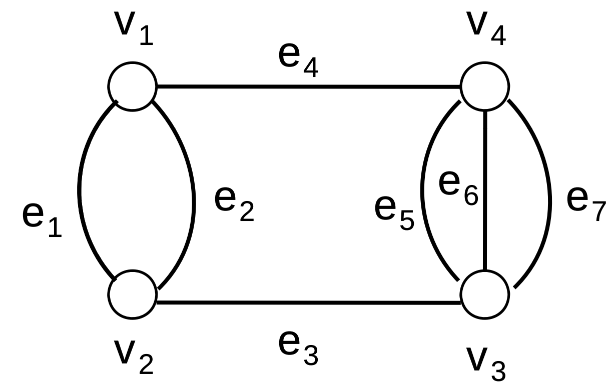
Image by MIT OpenCourseWare.



Degrees from A and B

- Given an adjacency matrix, A, can we determine $d(v_i)$?

- Can we determine $d(v_i)$ from incidence matrix, B?



$$A = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 3 \\ 1 & 0 & 3 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Complete Graphs

- What is the total number of edges, m, in a **complete graph**?
- For a graph of order n (i.e. n vertices), what is the number of total possible combinations if we pick two vertices at a time?
- Think *Combination – out of n objects, how many combinations are possible if we pick k objects at a time?*
- This is given by the Binomial coefficient:

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots 1} = \frac{n!}{k!(n-k)!}$$

$$k \leq n$$

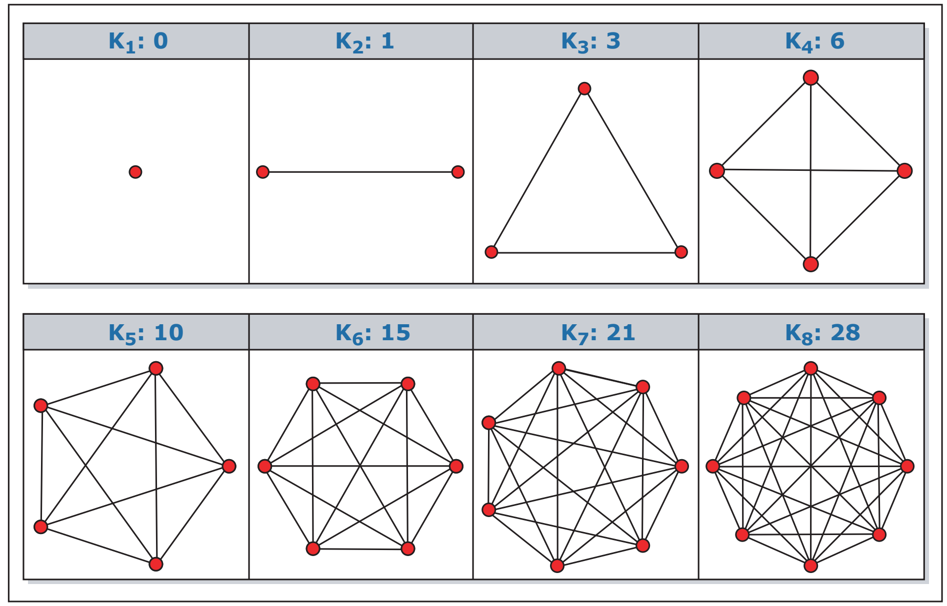


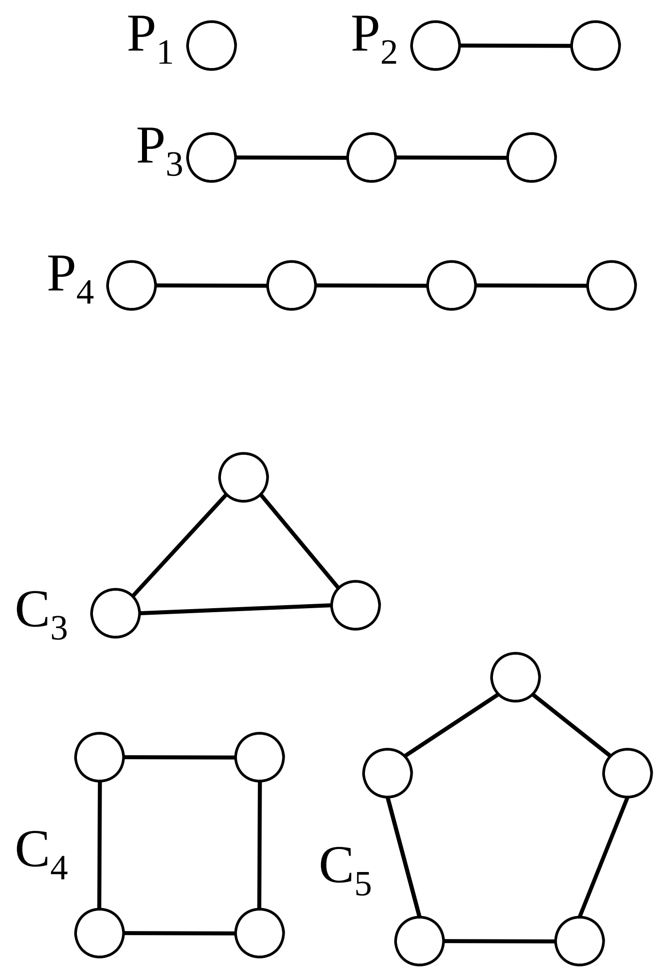
Image by MIT OpenCourseWare.

For a complete graph with order n, K_n, the size m is:

$$m = \binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)(n-2)!}{2!(n-2)!} = \boxed{\frac{n(n-1)}{2}}$$

Paths and Cycles

- Paths and cycles are two classes of graphs.
- For $n \geq 1$, the path P_n is a graph of order n and size n-1 whose vertices are $v_1, v_2, \ldots v_n$, and whose edges are $v_i v_{i+1}$ for i=1,…, n-1.
- For $n \geq 3$, the **cycle C_n is a graph of order n and size n** whose vertices can be labeled by $v_1, v_2, \ldots v_n$ and whose edges are $v_1 v_n$, and $v_i v_{i+1}$ for i=1,2,n-1. The cycle C_n is also referred to as an n-cycle.
- Note every cycle has vertices with the same number of degree: 2, and the number of edges in the cycle = number of vertices



Walks

- In a graph, we may wish to know if a route exists from one vertex to another- two vertices may not be adjacent, but maybe connected through a *sequence* of edges.
- A walk in a graph G is an alternating sequence of vertices and edges :

$$v_0\ e_0\ v_1\ e_1\ v_2 \ldots v_{k-1}\ e_{k-1}\ v_k$$

where $k \geq 1$ and e_i is incident with v_i and v_{i+1} for each $i = 0, 1, \ldots k-1$

- The vertices and edges need not be distinct
- The length of the walk W is defined as '*k*' which is the number of occurrences of edges in the sequence.

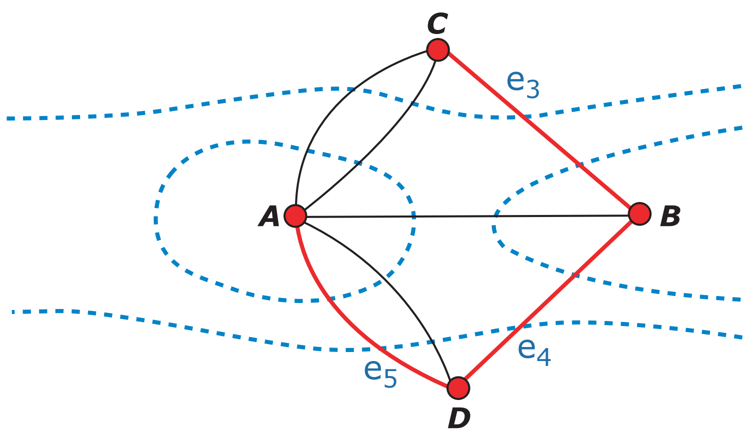


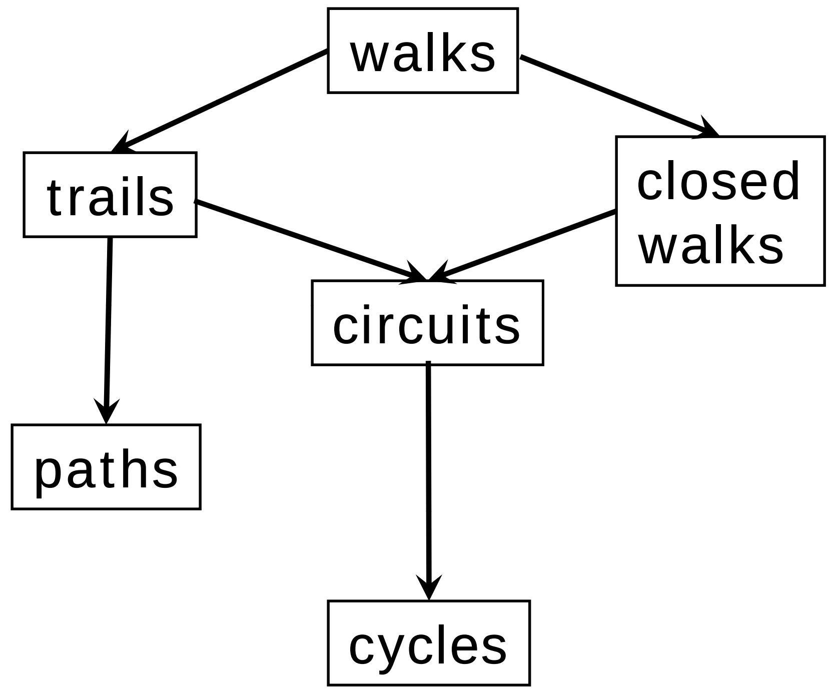
Image by MIT OpenCourseWare.

$C\ e_3\ B\ e_4\ D\ e_5\ A$

what is the length of this walk?

Trails and Circuits

- A *trail* in G is a walk with all of its edges e_1 e_2 …e_k distinct
- A path is G is a walk with all of its vertices v_0 v_1 …v_k distinct
- For vertices u and v in G, a u, v-walk (or trail, path etc.) is one with initial vertex u and final vertex v.
- A walk or trail of length at least 1 is closed if the initial and final vertex are the same. A closed trail is also called a circuit.
- A cycle is a closed walk with distinct vertices except for the initial and final vertex, which are the same.



Ref [4]

Examples

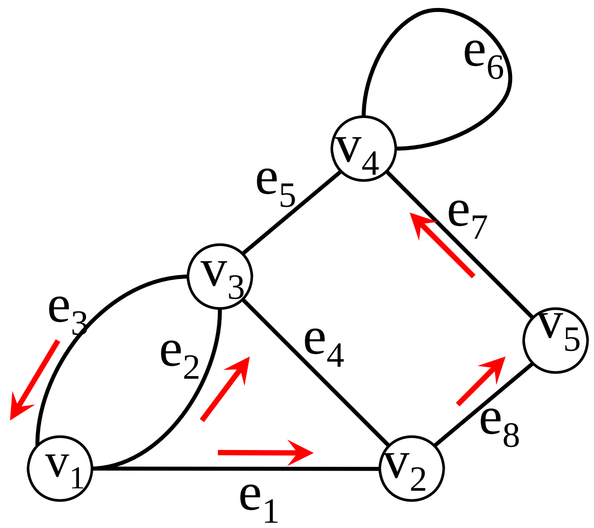
- The walk t is a trail of length 5:

  $t = (v_1, e_2, v_3, e_3, v_1, e_1, v_2, e_8, v_5, e_7, v_4)$

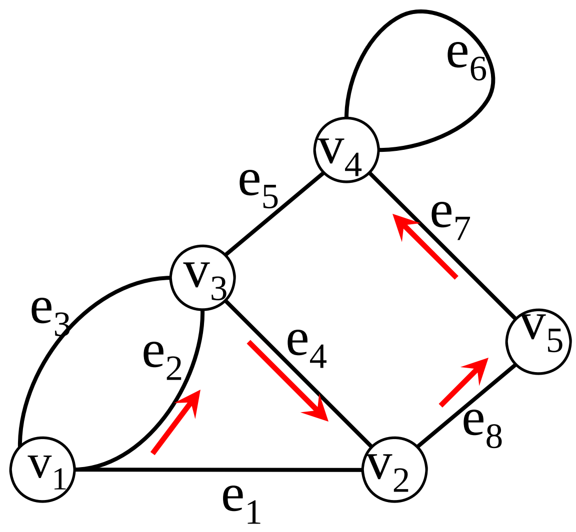
  t is not a path since v_1 appears twice

- The walk p is a path of length 4:

  $p = (v_1, e_2, v_3, e_4, v_2, e_8, v_5, e_7, v_4)$



Ref [4]

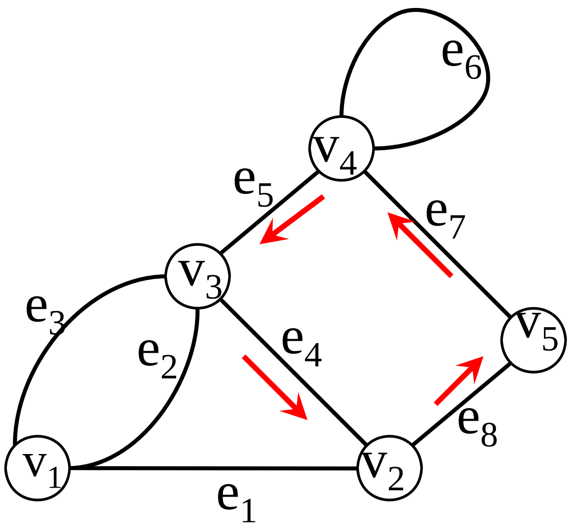


Ref [4]

Examples

- The walk *c* is a cycle of length 4:

  $c = (v_3, e_4, v_2, e_8, v_5, e_7, v_4, e_5, v_3)$



Ref [4]

Connectivity

- A graph G is **connected** if every two vertices in G are joined by a path.
- A graph is disconnected if it is not connected.
- A path in G that includes every vertex in G is called a **Hamiltonian path** of G.
- A cycle in G that includes every vertex in G is called a **Hamiltonian cycle** of G.
- If G contains a Hamiltonian cycle, then G is called a **Hamiltonian graph**.

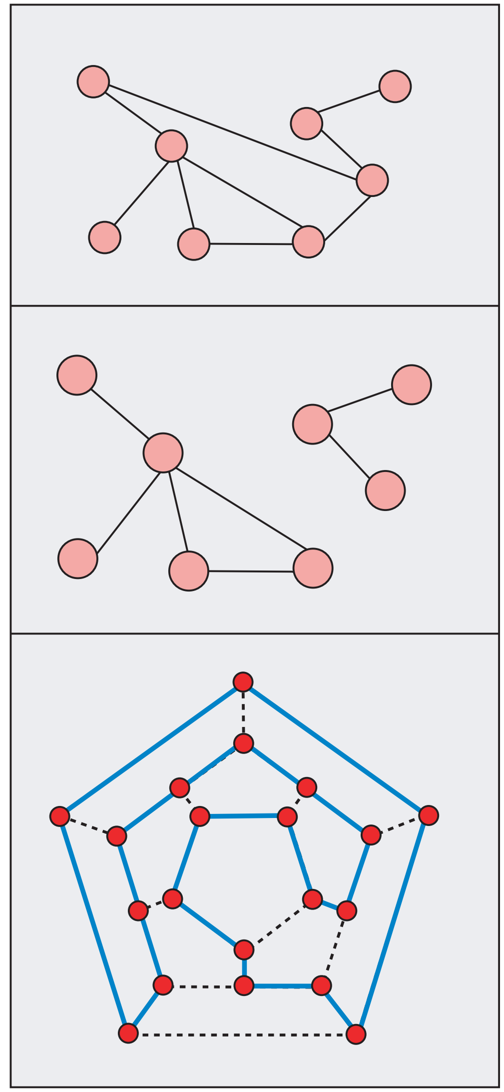


Image by MIT OpenCourseWare.

Trees

- A tree, *T*, is a connected graph that has no cycle as a subgraph
- A tree is a simple graph on *n* vertices- a tree cannot have any loops or multiple edges between two vertices.
- T has n-1 edges and is connected.
- A vertex v of a simple graph is called a leaf if d(v) = 1.
- Between every pair of distinct vertices in T there is exactly one path.
- Trees are useful in modeling applications such as hierarchy in a business, directories in an operating system, computer networks.

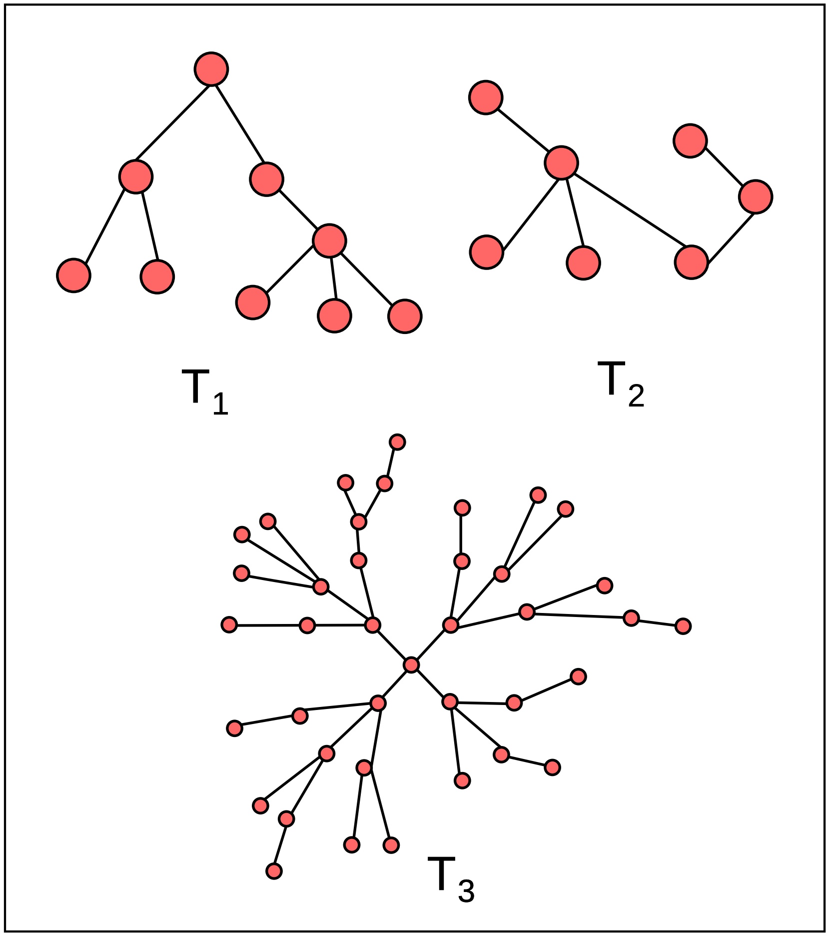


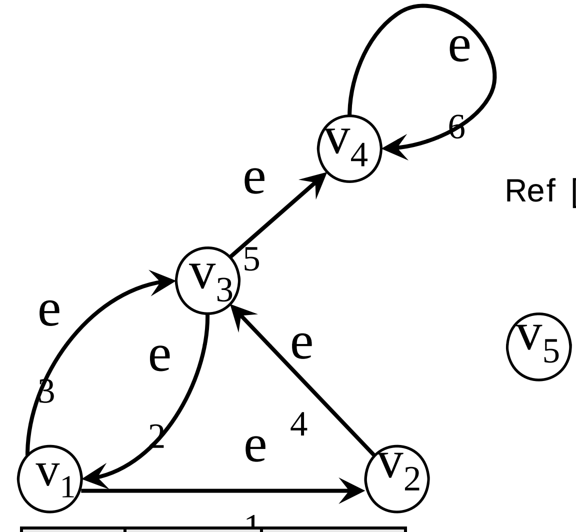
Image by MIT OpenCourseWare.

Directed Graphs

- A directed graph, or **digraph** G, consists of *directed* edges (represented with arrows).
- In a directed edge uv, the vertex u is called the *tail* and vertex v is called the *head* of the edge.

- The *indegree* $d^{-}(v)$ of a vertex v is number of directed edges having v as head.
- The *outdegree* $d^{+}(v)$of v is number of directed edges having v as tail.
- For a digraph:

$$\sum_{i=1}^{n} d^{-}(v_i) = \sum_{i=1}^{n} d^{+}(v_i) = m$$



Ref [4]

	$d^{-}(v_i)$	$d^{+}(v_i)$
v_1		
v_2		
v_3		
v_4		
v_5		

Weighted Graphs

- A connected graph G is called a **weighted graph** if each edge *e* in G is assigned a number $w(e)$, called the weight of e.
- Depending on the application, the weight of an edge may be a measure of physical distance, time consumed, cost, capacity, or some other quantity of interest.
- Given a walk W in a weighted graph, the weight of W, is the sum of the weights of the edges contained in W.

Traveling Salesman Problem

A traveling salesman wants to make a round trip through n cities, $c_1..c_i..c_n$. He starts in c_1, visits each remaining city c_i exactly once, and ends in c_1 where he started the trip.

If he knows the distances between every pair of cities c_i and c_j, how should he plan his round trip to make the total round-trip distance as short as possible?

The problem of finding the shortest route is that of finding a minimum weight Hamiltonian cycle of the weighted complete graph K_n.

Application Example: Project Graphs and Critical Paths

- A project consists of a collection of tasks.
- Each task has an associated completion time.
- A task may depend on other tasks to be completed before it can be initiated.
- A project graph can be constructed, such that the vertices represent tasks, and edges represent task dependencies.
- The total time of a path is the sum of completion time of each task on that path.

- **The path with longest total time is the critical path**
- The critical path determines project completion time.

Job #	Immediate Predecessors	Time [min]
A		0
B	A	10
C	A	20
D	B,C	30
E	B,C	20
F	E	40
G	D,F	20
H	G	0

k-regular graphs

- A graph g is called k-regular if $d(v_i) = k$ for all v_i in G.
- The null graph is a 0-regular graph.
- The cycle C_n is a 2-regular graph.
- A complete graph is an (n-1) regular graph.

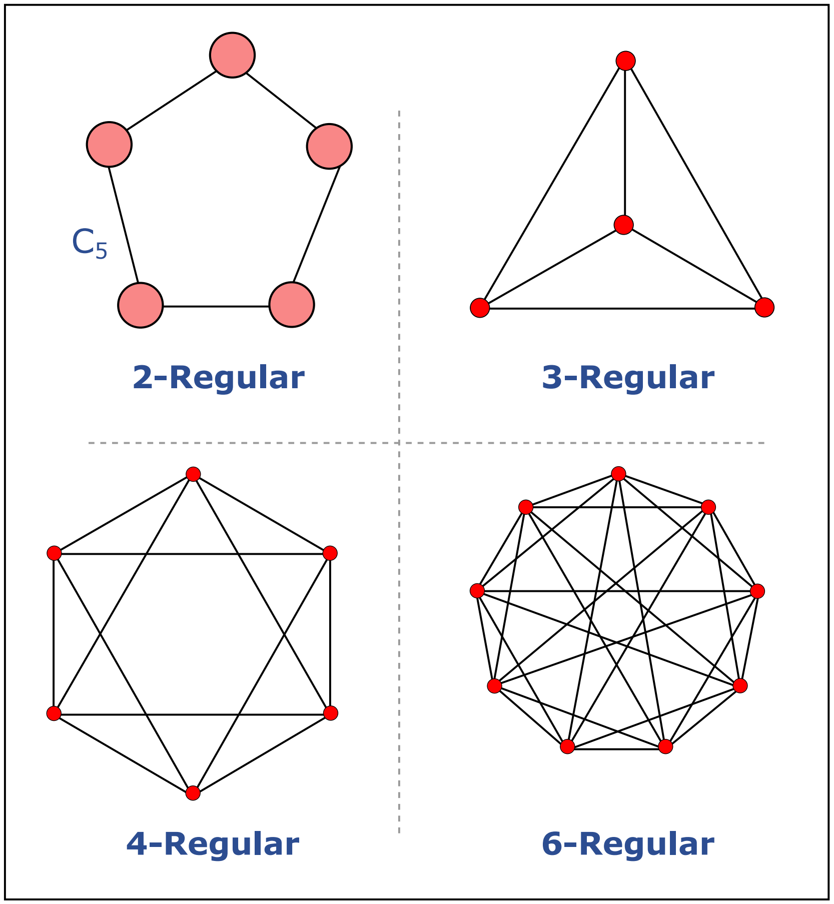
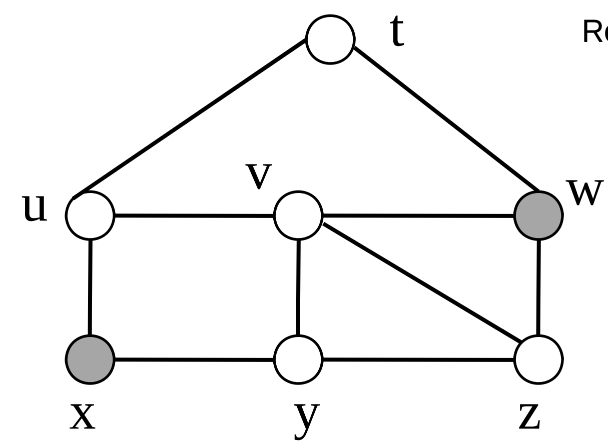


Image by MIT OpenCourseWare.

Distance

- **The distance** from vertix v_1 to v_2, $d(v_1, v_2)$ in a connected graph G **is the smallest length of all v_1 - v_2 paths** in G.
- The shortest path through the network from one vertex to another is also called the '**geodesic path**'.
- There maybe and often is more than one geodesic path between two vertices.

Ref [3]



x-w path	length

$d(x, w) =$

Diameter

- The greatest distance (longest path) between any two vertices in a graph G is called the **diameter** of G.
- The diameter is the **longest geodesic path** between any two vertices in the network.
- The diameter of a graph is an indication of how far apart are its vertices.

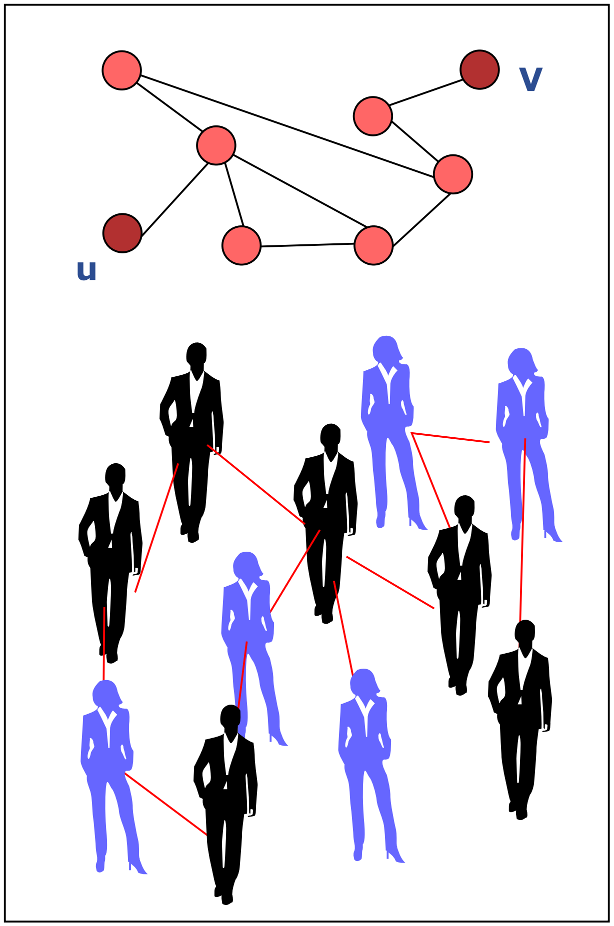


Image by MIT OpenCourseWare.

Diameter of our World

- We may model our world as a collection of people – each person is a vertex (node) and two people (vertices) are connected if they are acquainted. What will be the diameter of this graph (or social network)?

Small World Networks

- Small world networks are 'highly clustered', yet have small characteristic path lengths.
- Neural networks, power grids, collaboration graphs of film actors, and many other systems form 'small world' networks.
- In small world networks there are 'short cuts' that shorten the distance between vertices.
- Signal propagation speed is enhanced in such systems; rumors can spread quickly, the number of legs in an air or train journey is small, infectious diseases spread more easily in a population etc.

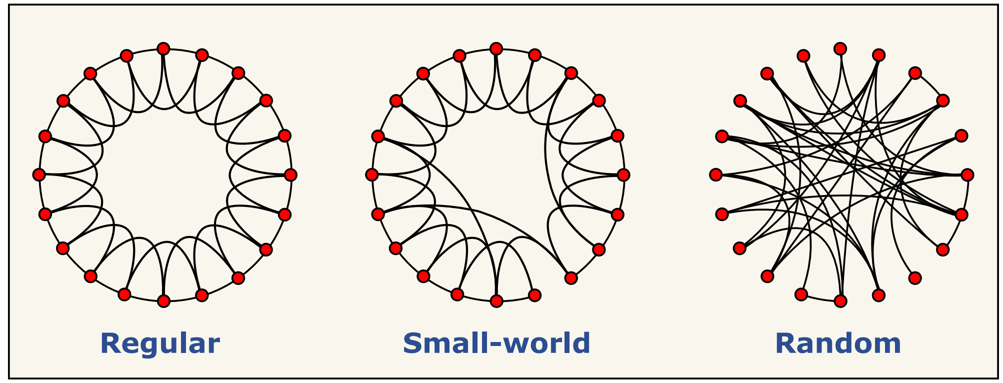


Image by MIT OpenCourseWare.

Duncan J. Watts & Steven H. Strogatz, "Collective dynamics of 'small world' networks', Nature, Vol. 393, 4 June 1998

References

[1]M.E.J. Newman, “The structure and function of complex networks”, SIAM review, 2003

[2] Duncan J. Watts & Steven H. Strogatz, “Collective dynamics of ‘small world’ networks’, Nature, Vol. 393, 4 June 1998

[3] Introduction to Graph Theory, Koh Khee Meng et. al

[4] Graph Theory: Modeling, Applications, and Algorithms, Geir Agnarsson, Raymond Greenlaw

ESD.00 Introduction to Engineering Systems
Spring 2011