Graph Theory

Dr/ Hanan Hamed

Lecture1

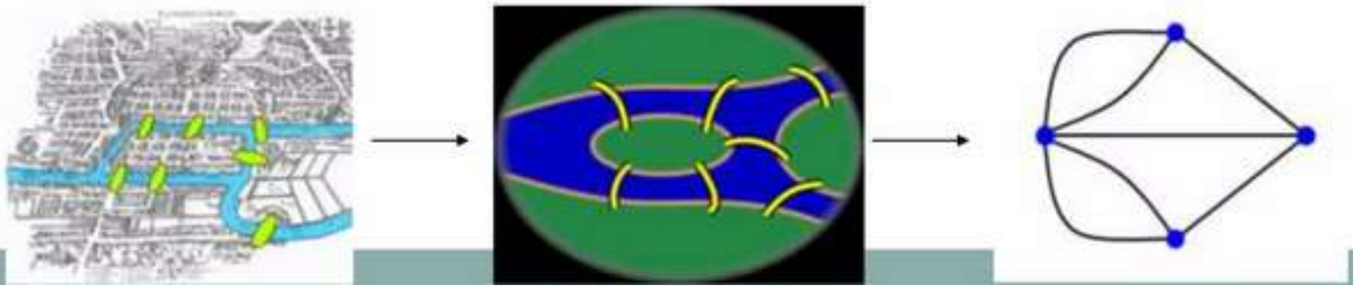
Why Graph Theory ?

- Graphs used to model pair wise relations between objects
- Generally a network can be represented by a graph
- Many practical problems can be easily represented in terms of graph theory

Graph Theory - History

- Begun in 1735
- Mentioned in Leonhard Euler's paper on "*Seven Bridges of Konigsberg* ".

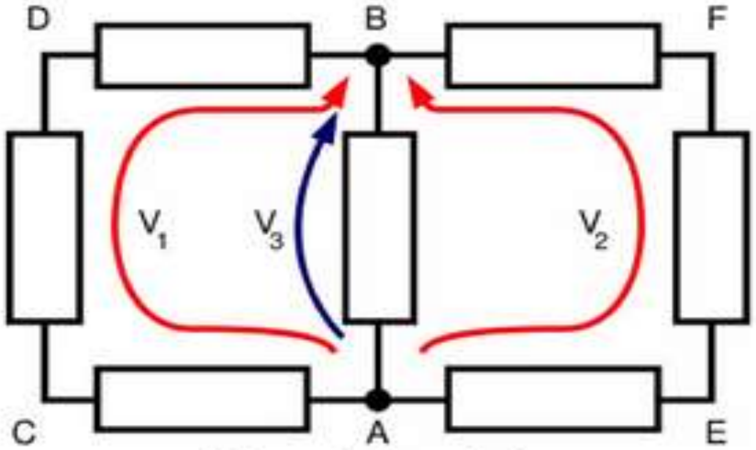
Problem : Walk all 7 bridges without crossing a bridge twice



Graph Theory – History.....

Trees in Electric Circuits

Gustav Kirchhoff



$$V_1 = V_2 = V_3$$

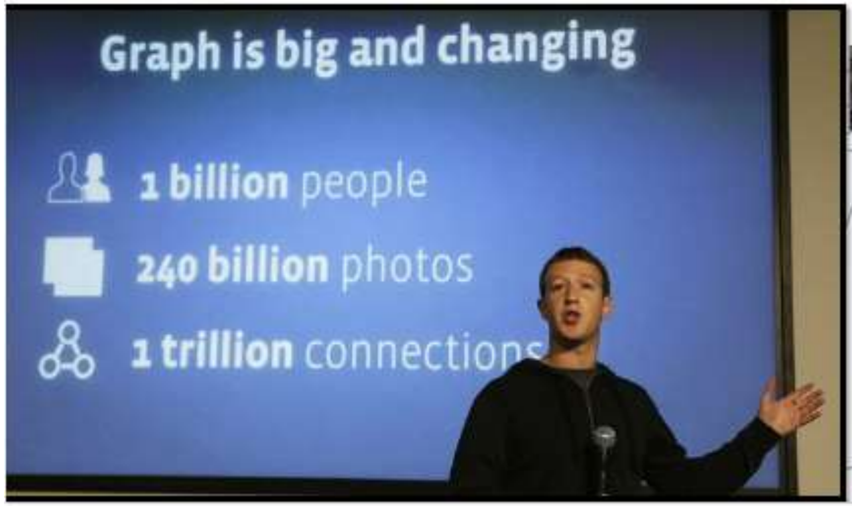
Graph Theory – History.......

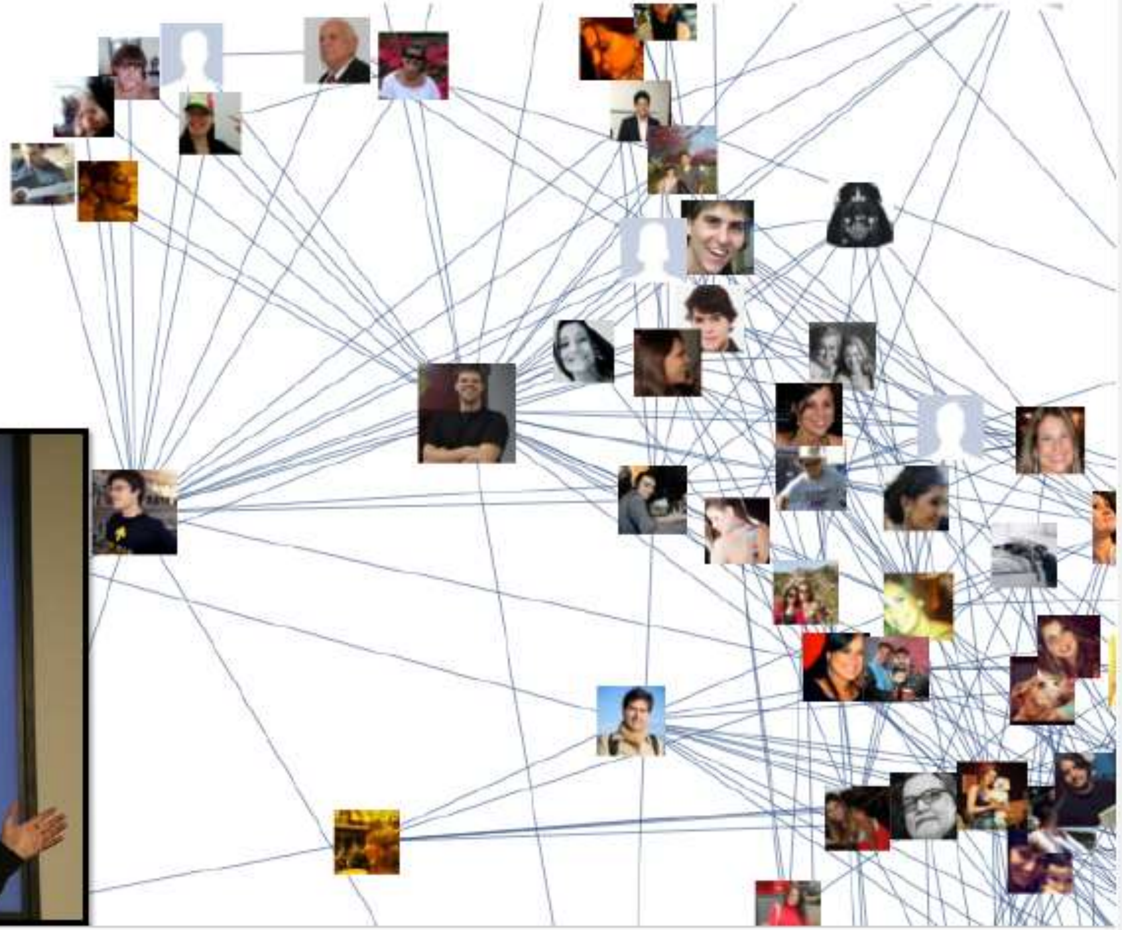
Cycles in Polyhedra - polyhedron with no Hamiltonian cycle

Thomas P. Kirkman

William R. Hamilton

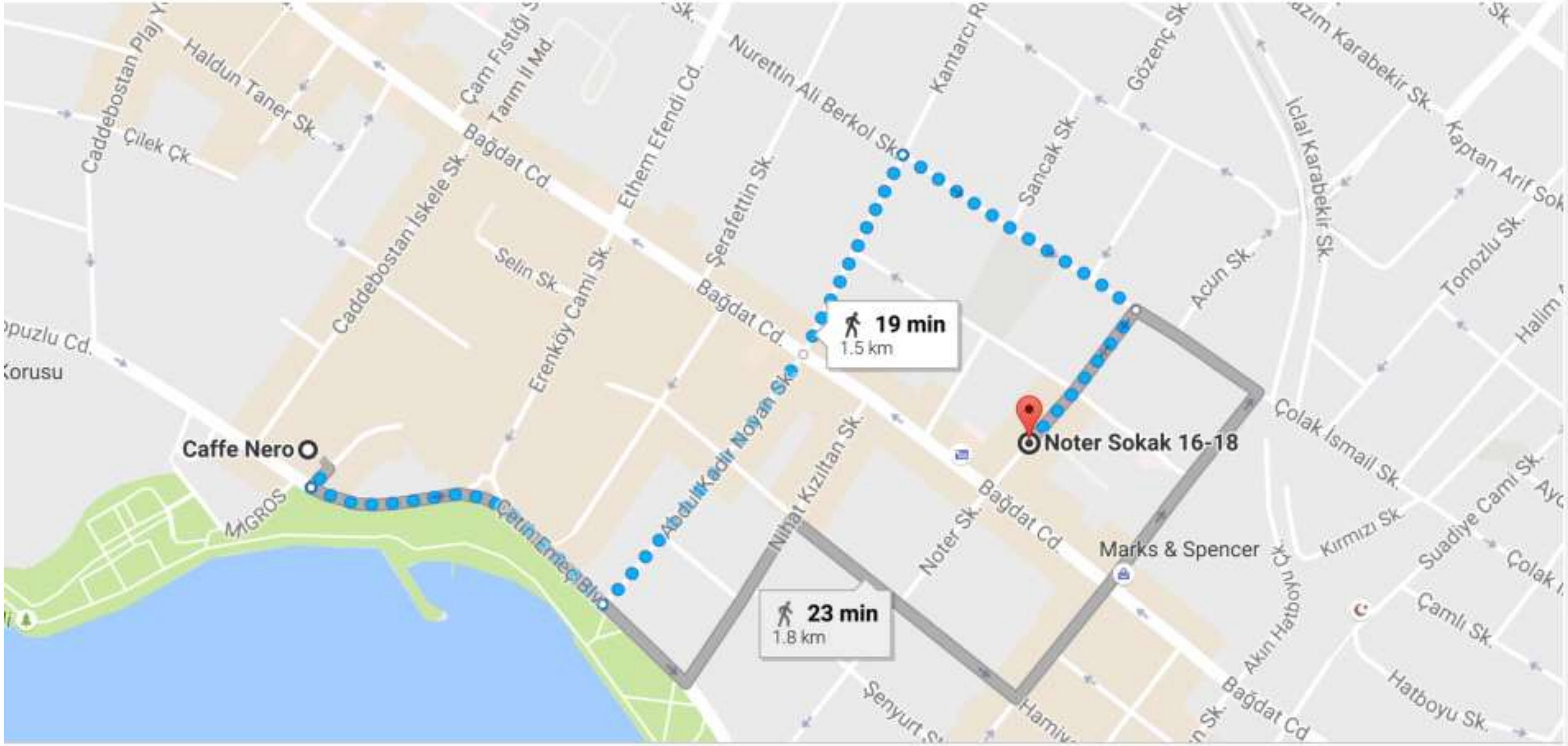
Facebook


Graph is big and changing
1 billion people
240 billion photos
1 trillion connections



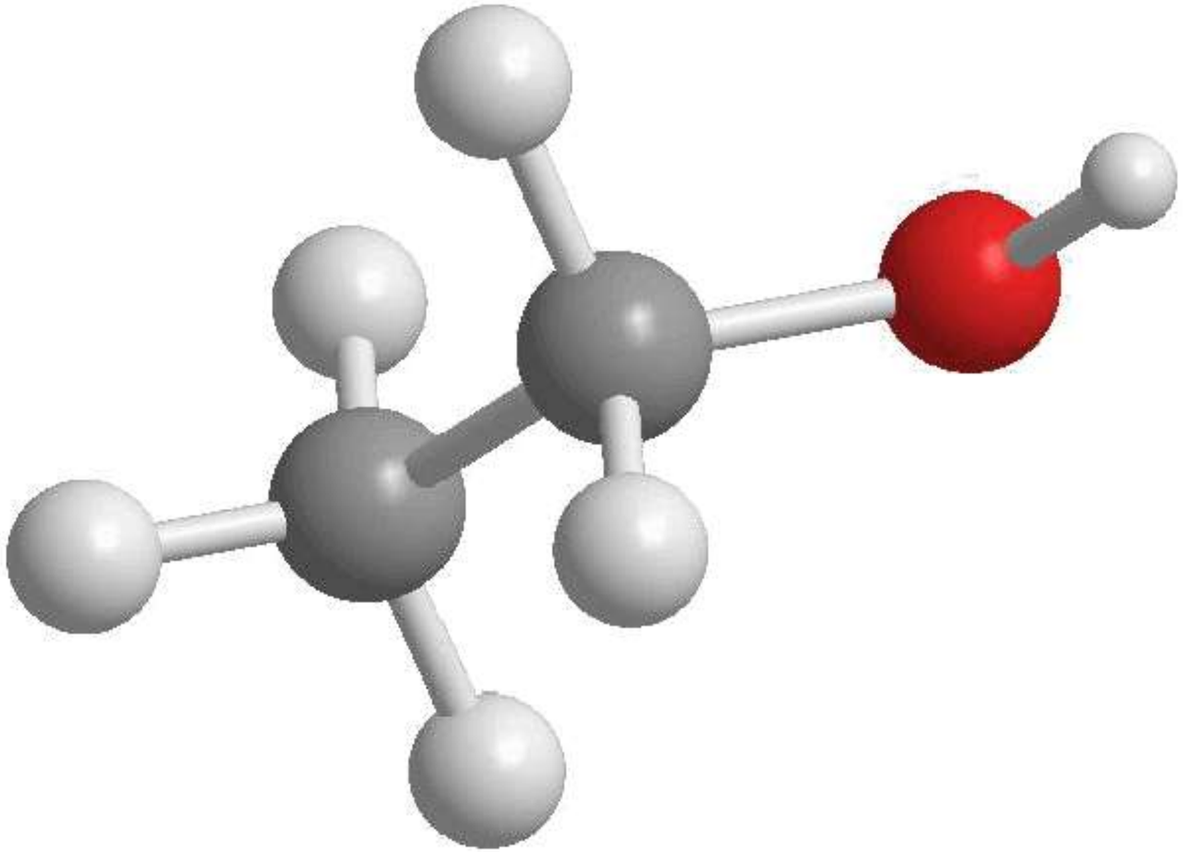

facebook
Me too!

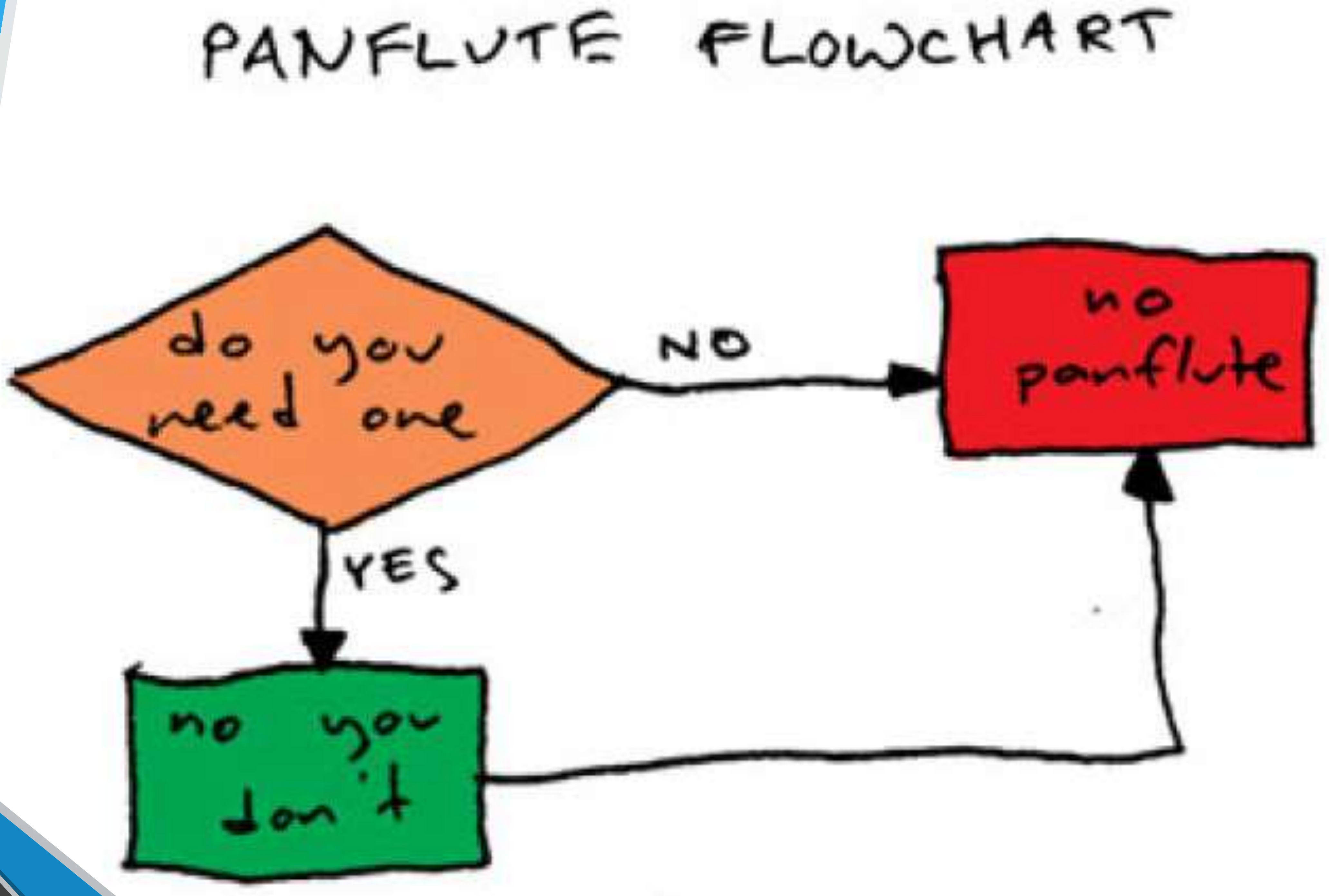
Google Maps

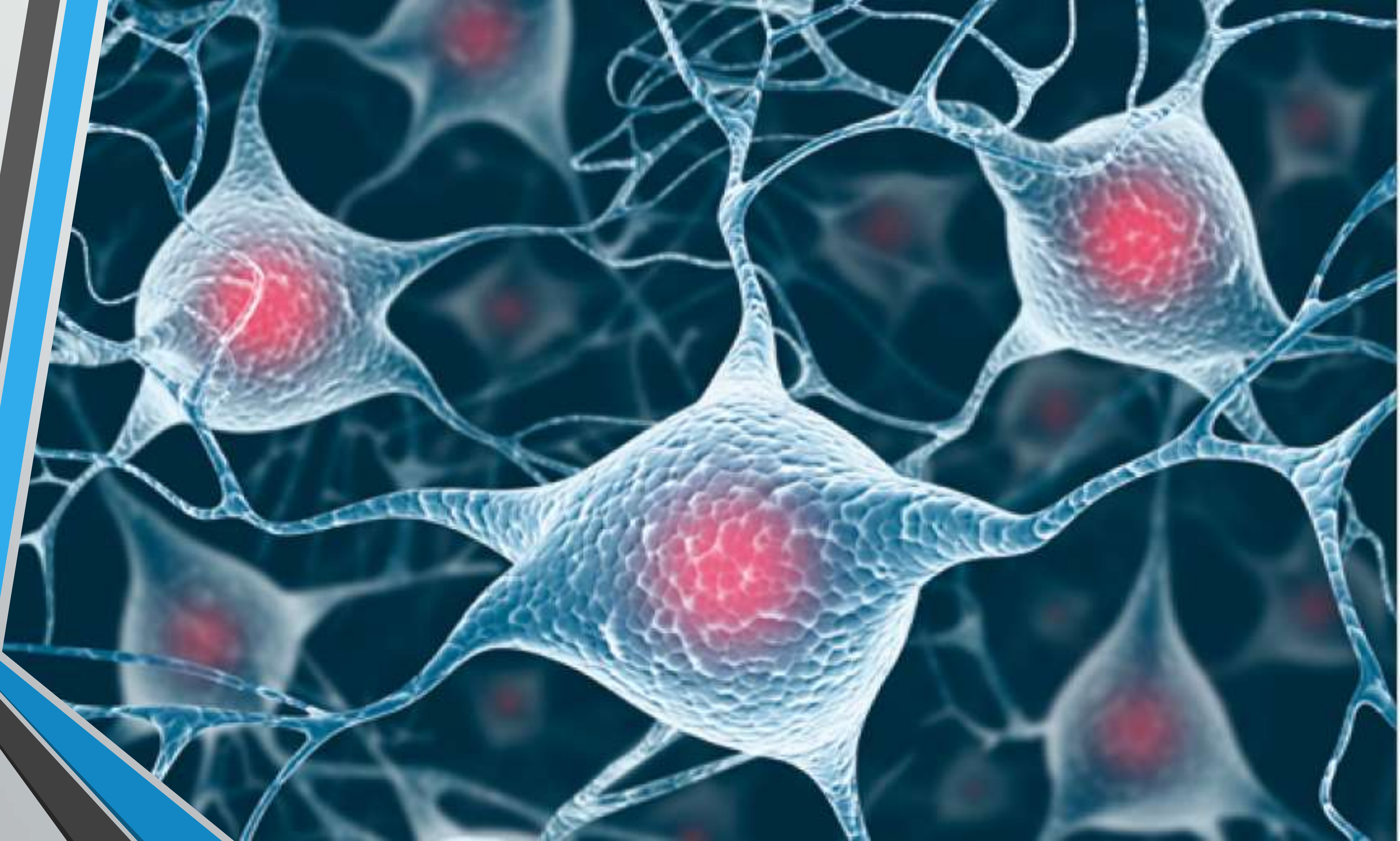


Caffe Nero
Noter Sokak 16-18
19 min
1.5 km
23 min
1.8 km
Bağdat Cd.
Marks & Spencer
Haldun Taner Sk.
Caddebostan Plaj Y.
Çilek Çk.
Caddebostan İskele Sk.
Selin Sk.
Erenköy Cami Sk.
Ethem Efendi Cd.
Şerafettin Sk.
Nurettin Ali Berkol Sk.
Kantarcı R
Sancak Sk.
Gözenç Sk.
Acun Sk.
İclal Karabekir Sk.
Kaptan Arif Sok
Tonozlu Sk.
Çolak İsmail Sk.
Kırmızı Sk.
Suadiye Cami Sk.
Çamlı Sk.
Hatboyu Sk.
Akın Hatboyu Çk.
Nihat Kızıltan Sk.
Noter Sk.
Şenyurt S
Çetin Emeç Blv.
Abdulkadir Noyan Sk.
MIGROS
Korusu

Chemical Bonds

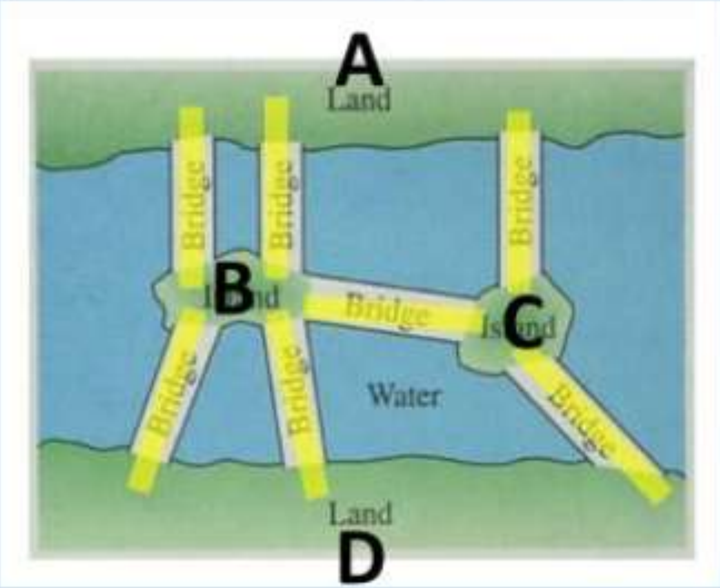


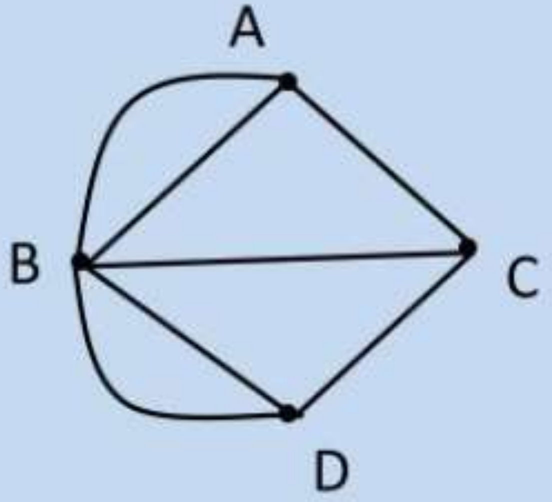

PANFLUTE FLOWCHART
do you need one
NO
no panflute
YES
no you don't



thedigitalbridges.com/wp-content/uploads/2015/11/internet-small-world.jpg

Ex) Represent the "Konigsberg Bridge" problem using a vertex-edge graph.



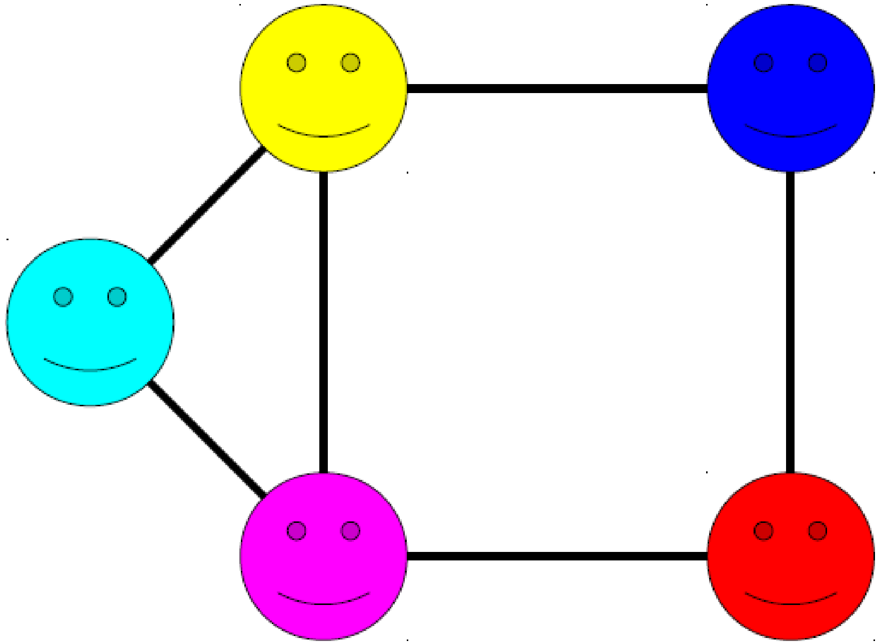


* Vertices represent locations.
* Edges represent "connections" between those locations.

What's in Common

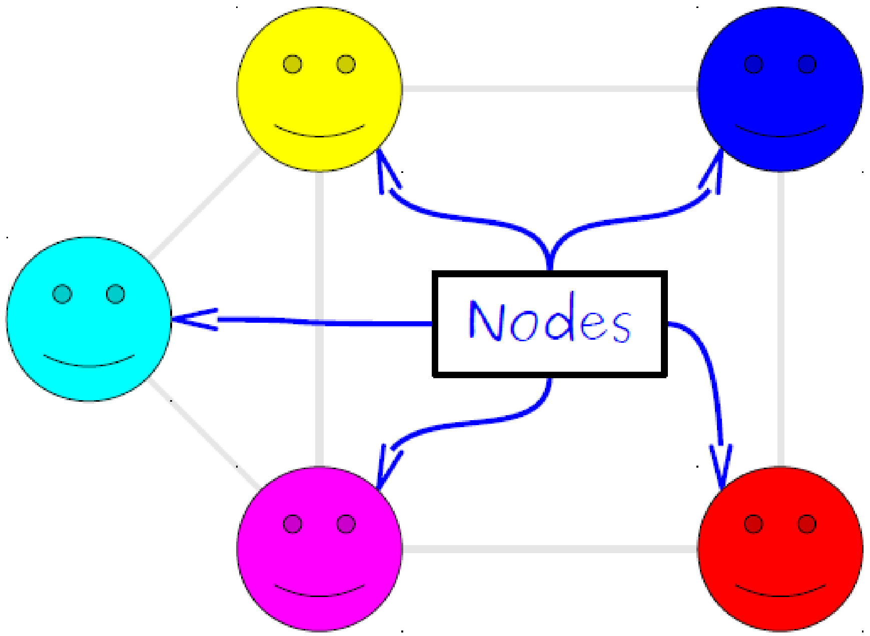
- Each of these structures consists of
  - a collection of objects and
  - links between those objects.
- ***Goal:*** find a general framework for describing these objects and their properties.

A ***graph*** is a mathematical structure for representing relationships.



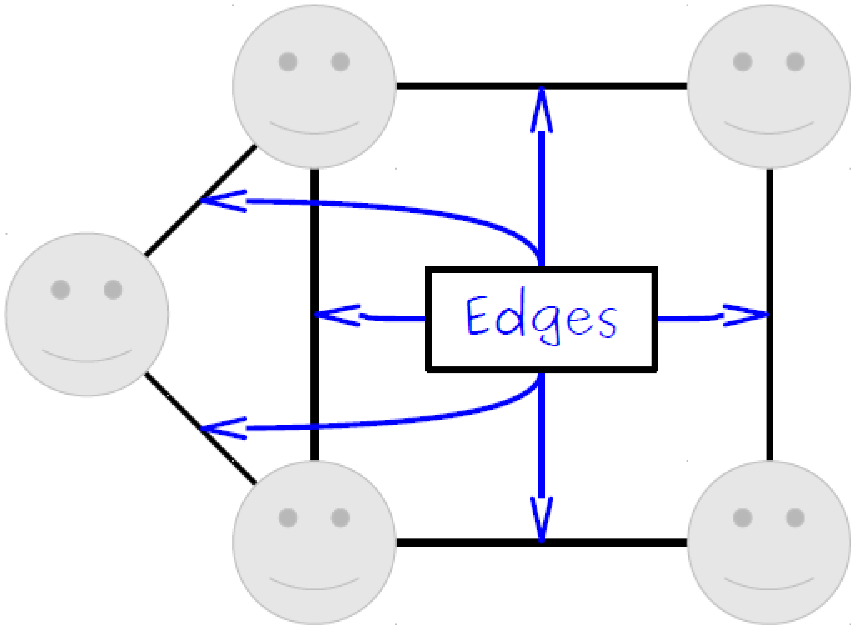
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

A ***graph*** is a mathematical structure for representing relationships.



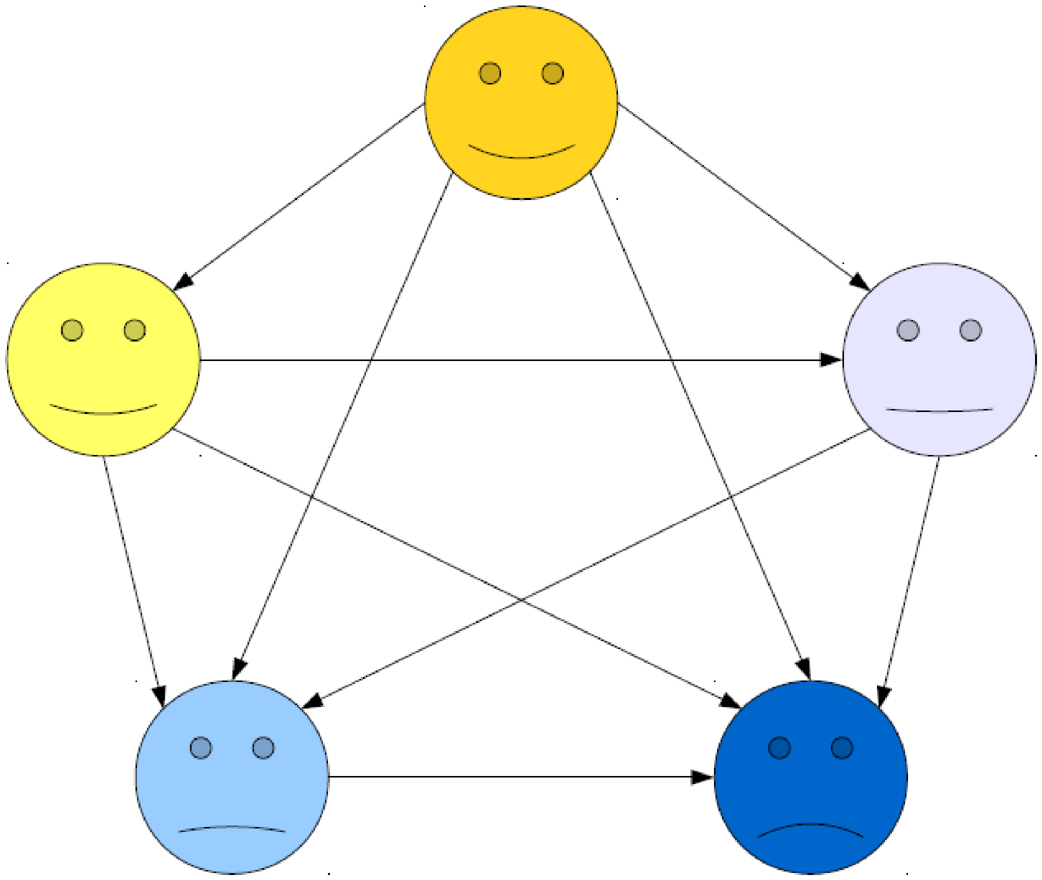
A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

A ***graph*** is a mathematical structure for representing relationships.

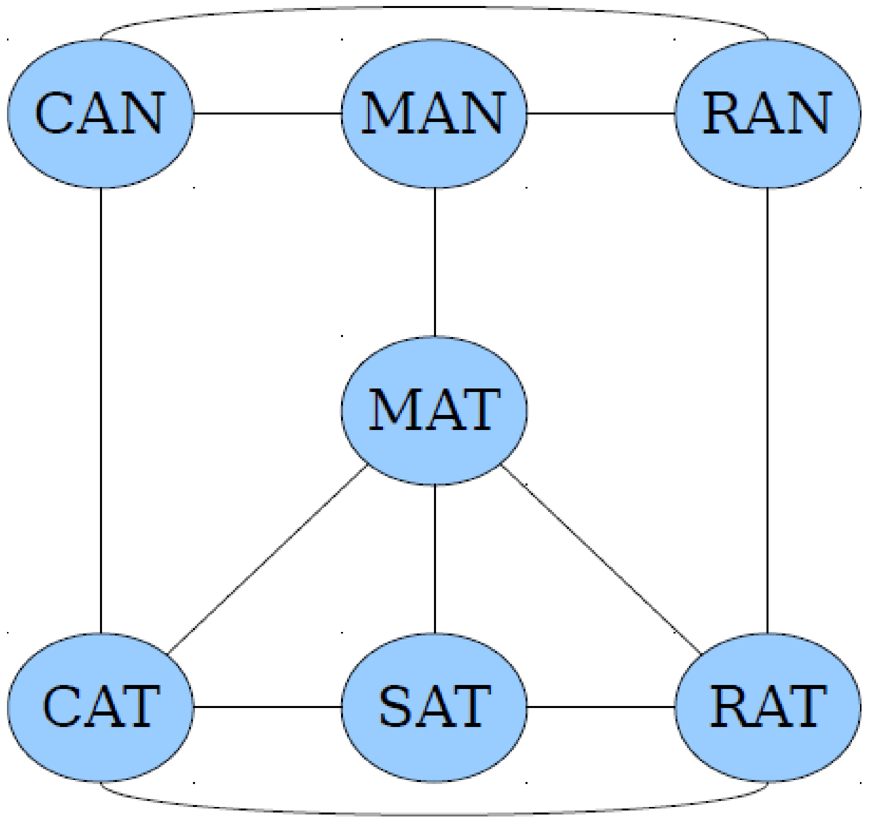


A graph consists of a set of ***nodes*** (or ***vertices***) connected by ***edges*** (or ***arcs***)

Some graphs are ***directed***.



Some graphs are ***undirected***.



Going forward, we're primarily going to focus on undirected graphs.

The term “graph” generally refers to undirected graphs unless specified otherwise.

Formalizing Graphs

- How might we define a graph mathematically?
- We need to specify
  - what the nodes in the graph are, and
  - which edges are in the graph.
- The nodes can be pretty much anything.
- What about the edges?

Graph Theory

- Graphs can be used to model many types of relations and process dynamics in physical, biological, social and information systems
- Graphs can be used to represent networks of communication, data organization, computational devices, the flow of computation, **the link structure of a website**, to study molecules in chemistry and physics, **friends**, etc.
- Many practical (**and lucrative**) problems can be represented by graphs

INTRODUCTION

- What is a graph G?

It is a pair $G = (V, E)$, where

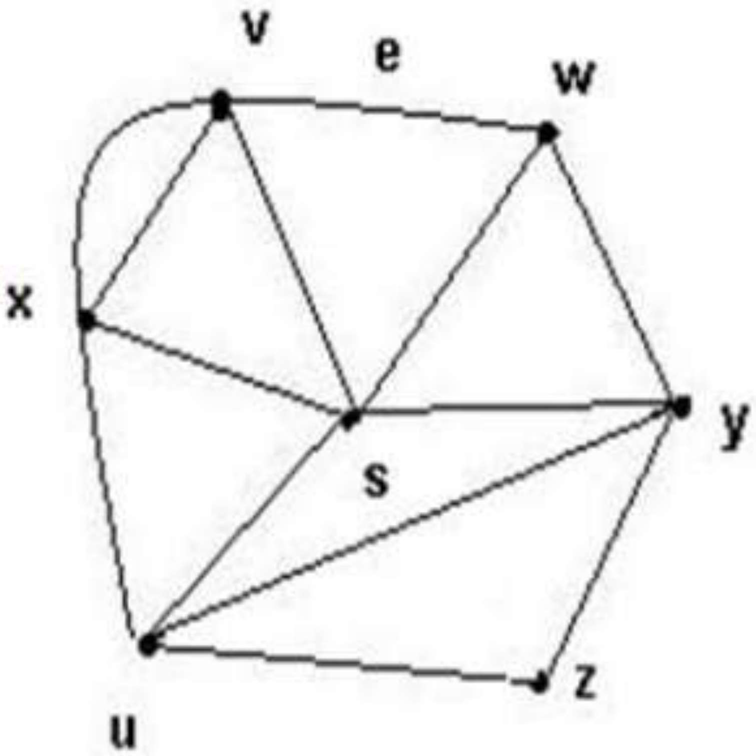
$V = V(G)$ = set of vertices

$E = E(G)$ = set of edges

- **Example:**

$V = \{s, u, v, w, x, y, z\}$

$E = \{(x,s), (x,v)_1, (x,v)_2, (x,u), (v,w),$
$(s,v), (s,u), (s,w), (s,y), (w,y), (u,y), (u,z), (y,z)\}$



1. Point

A **point** is a particular position that is located in a space. Space can be one-dimensional, two-dimensional or three-dimensional space. A dot is used to represent a point in graph and it is labeled by alphabet, numbers or alphanumeric values.

Example

• **p**

Here, dot is a point labeled by 'p'.

2. Line

Two points are connected to each other through a **line**. A **line** is a connection between two points. It is represented by a solid line.

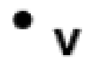
Example

3. Vertex

A **vertex** is a synonym of point in graph i.e. one of the points on which the graph is defined and which may be connected by lines/edges is called a vertex.

Vertex is also called "node", "point" or "junction". A vertex is denoted by alphabets, numbers or alphanumeric value.

Example



Here, point is the vertex labeled with an alphabet 'v'.

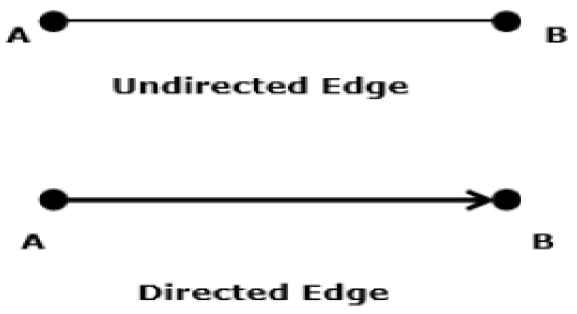
4. Edge

Edge is the connection between two vertices. Each edge connects one vertex to another vertex in the graph. Without a vertex, an edge cannot be formed. It is also called line, branch, link or arc.

Edge can either be **directed** or **undirected**. A directed edge is the edge which points from one vertex to another, and an undirected edge has no direction.

If there is a directed edge from vertex A to B, and a directed edge from B to A, this would essentially be equivalent to an undirected edge connecting A and B.

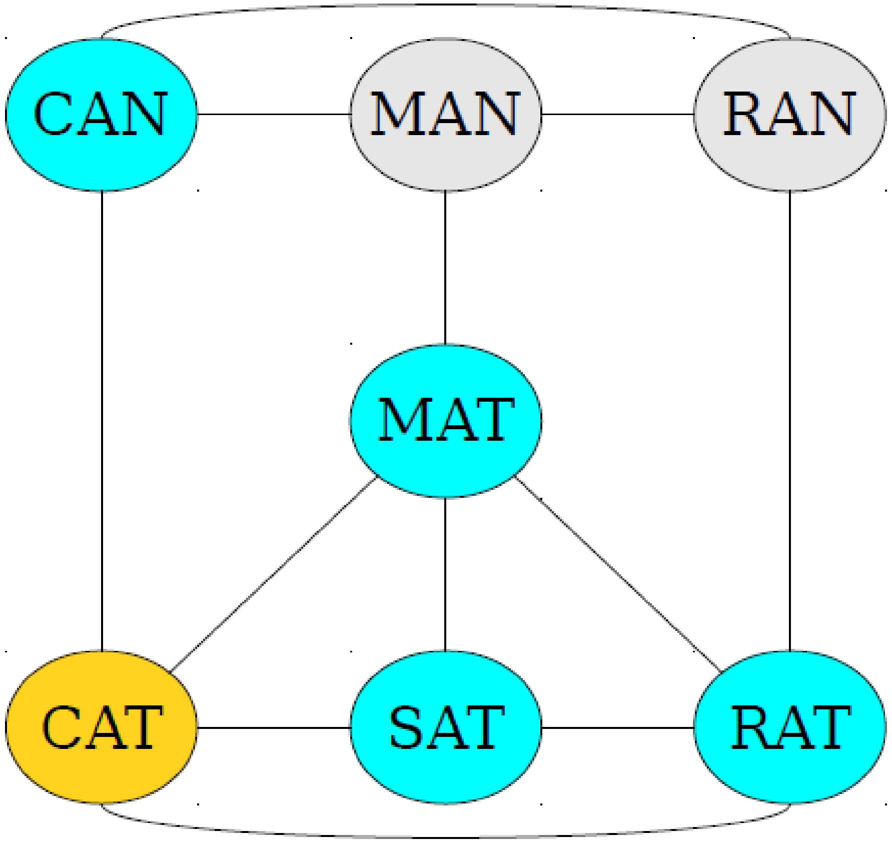
Example



Here, **'A' and 'B'** are the **vertices** and the link 'AB' between them is called an **edge**.

Formalizing Graphs

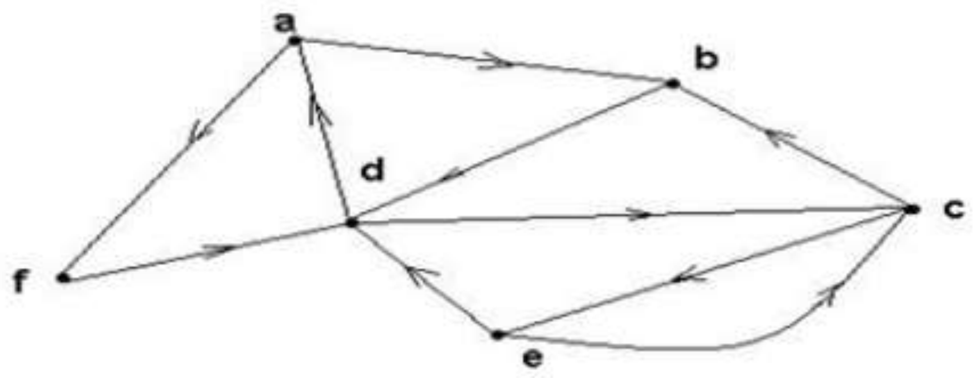
- An ***unordered pair*** is a set $\{a, b\}$ of two elements (remember that sets are unordered).
  - $\{0, 1\} = \{1, 0\}$
- An ***undirected graph*** is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are unordered pairs of nodes drawn from V.
- A ***directed graph*** is an ordered pair $G = (V, E)$, where
  - V is a set of nodes, which can be anything, and
  - E is a set of edges, which are *ordered* pairs of nodes drawn from V.



Two nodes are called ***adjacent*** if there is an edge between them.

Directed graphs (digraphs)

G is a *directed graph* or *digraph* if each edge has been associated with an ordered pair of vertices, i.e. each edge has a direction

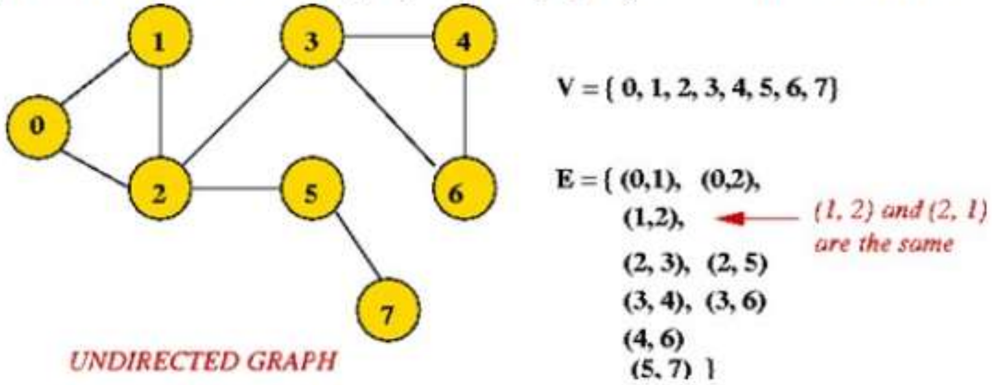


UNDIRECTED GRAPH

- Edges have no direction.
- If an edge connects vertices *1* and *2*, either convention can be used:

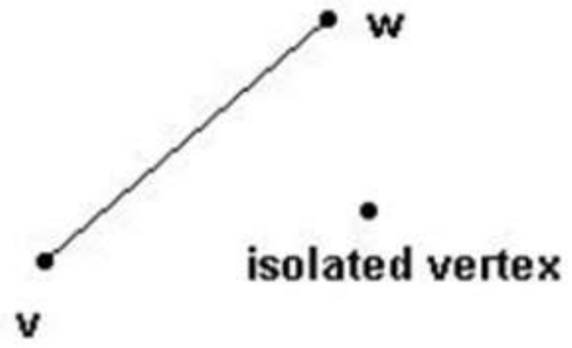
  No duplication: only one of *(1, 2)* or *(2, 1)* is allowed in *E*.

  Full duplication: both *(1, 2)* and *(2, 1)* should be in *E*.



Edges

- An edge may be labeled by a pair of vertices, for instance e = (v,w).
- e is said to be *incident* on v and w.
- Isolated vertex = a vertex without incident edges.

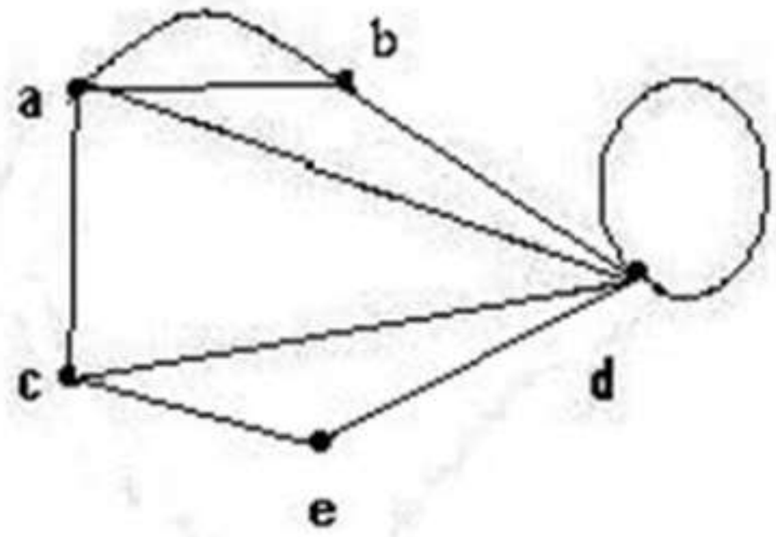


Special edges

- **Parallel edges**
  - Two or more edges joining a pair of vertices

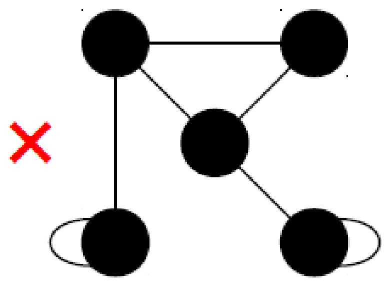
    in the example, **a** and **b** are joined by two parallel edges
- **Loops**
  - An edge that starts and ends at the same vertex

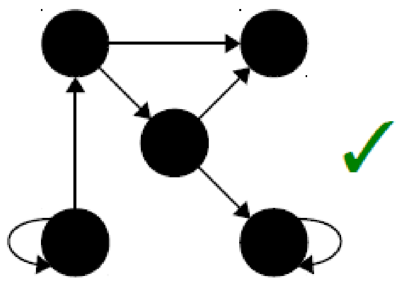
    In the example, vertex **d** has a loop



Self-Loops

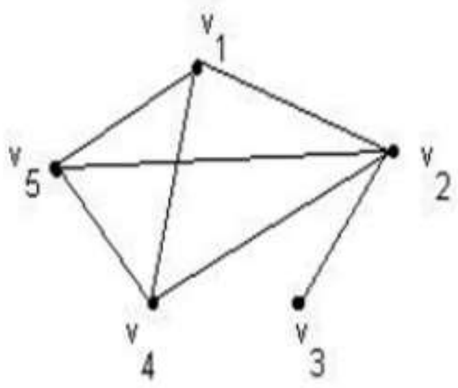
- An edge from a node to itself is called a ***self-loop***.
- In undirected graphs, self-loops are generally not allowed unless specified otherwise.
  - This is mostly to keep the math easier. If you allow self-loops, a lot of results get messier and harder to state.
- In directed graphs, self-loops are generally allowed unless specified otherwise.





Special graphs

- **Simple graph**
  - A graph without loops or parallel edges.



- **Weighted graph**
  - A graph where each edge is assigned a numerical label or "weight".