7.1 Introduction to graph theory: basic concepts

Notebook:	Discrete Mathematics [CM1020]		
Created:	2019-10-07 2:31 PM	**Updated:**	2019-12-12 1:04 PM
Author:	SUKHJIT MANN		

Cornell Notes	**Topic:** 7.1 Introduction to graph theory: basic concepts	Course: BSc Computer Science
		Class: Discrete Mathematics-Lecture
		Date: December 11, 2019

Essential Question:

What is a graph and how it is represented with edges, vertices, loops and paths?

Questions/Cues:

- What is a graph?
- What are the origins of graph theory?
- What are some real-world applications of graph theory?
- What is a more formal definition of a graph?
- What is a vertex?
- What is an edge?
- What is meant by adjacency in graph theory?
- What are loops and parallel edges?
- What is a directed graph or Digraph?
- What is a walk?
- What is a trail?
- What is a circuit?
- What is a path?
- What is a cycle?
- What is an Euler path?
- What is an Hamiltonian path?
- What is an Hamiltonian cycle?
- What is an Hamiltonian graph?
- What is connectivity in terms of graphs?
- What is strong connectivity?
- What is Transitive Closure?
- What is the degree of a vertex in terms of an undirected graph?
- What is the in-degree/out-degree of a vertex?
- What is the degree sequence of a graph?
- What are the properties of a degree sequence?
- What is a simple graph?
- What are the properties of simple graphs?
- What is a regular graph?
- What are the properties of a regular graph?
- What are some special regular graphs?
- What is a complete graph?

- What are the properties of a complete graph?

Notes

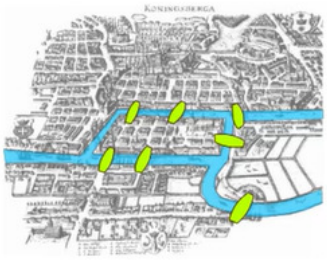
What is a graph?

Graphs are **discrete** structures consisting of **vertices (nodes)** and **edges** connecting them

Graph theory is an area in discrete mathematics which studies these type of discrete structures.

Origins of graph theory

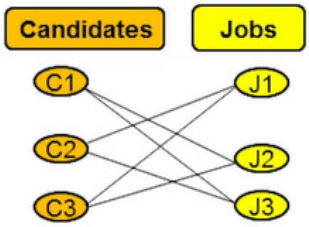
The first problem in graph theory is the **Seven Bridges** of Konigsberg problem solved by Leonhard Euler in 1735



Application of graphs

In a variety of disciplines, problems can be solved using graph models:

- Modelling computer networks
- Modelling road maps
- Solving shortest path problems between cities
- Assigning jobs to employees in an organisation.



Definition: Graph

G is an ordered triple G:=(V, E).

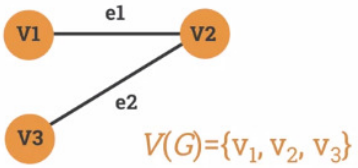
V is a set of nodes, or vertices.

E is a set of edges, lines or connections.

Definition: Vertex

Vertex

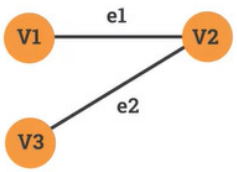
- Basic Element of a graph
- Drawn as a *node* or a *dot*
- Set of vertices of *G* is usually denoted by *V*(*G*) or *V*.



Definition: Edges

Edge

- A is a link between 2 vertices
- Drawn as a line connecting two vertices
- The set of edges in a graph G is usually denoted by E(G), or E.



$E(G)=\{e_1, e_2\} = \{\{v_1, v_2\}, \{v_2, v_3\}\}$

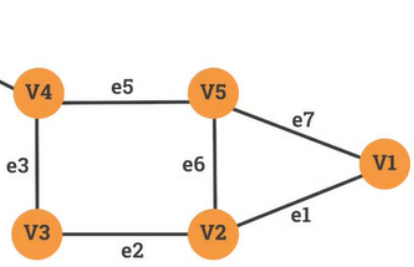
Definition: Adjacency

Adjacency

- Two **vertices** are said to be **adjacent** if they are endpoints of the same edge
- Two **edges** are said **adjacent** if they share the same vertex
- If a vertex **v** is an **endpoint** of an edge **e**, then we say that **e** and **v** are **incident**.

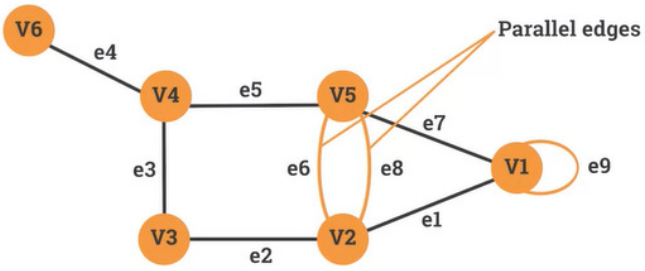
Example

- $V:=\{v_1, v_2, v_3, v_4, v_5, v_6\}$
- $E=\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$



v_1 and v_2 are endpoints of the edge e_1. We say that v_1 and v_2 are **adjacent**.
The edges e_1 and e_7 share the same vertex v_1. We say that e_1 and e_7 are **adjacent**.
The vertex v2 is an endpoint of the edge e_1. We say that e_1 and v_2 are **incident**.

Loops and parallel edges



v_2 and v_5 are are linked with two edges (e_6 and e_8).
e_6 and e_8 are called parallel edges.

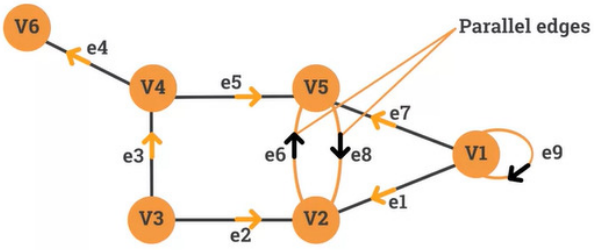
v_1 is linked to itself by e_9. The edge e_9 is called a loop.

Directed graphs — Digraph

A **directed graph**, also called a **digraph**, is a graph in which the edges have a direction.

This is usually indicated with an arrow on the edge.

Directed graphs



e_1 is a connection from v_1 to v_2 but not from v_2 to v_1

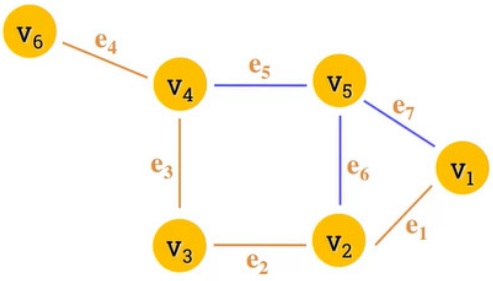
e_6 is a connection from v_2 to v_5 whereas e_8 is a connection from v_5 to v_2

Definition of a walk

A walk is a sequence of vertices and edges of a graph were vertices and edges can be repeated.

A **walk of length k** in a graph is a succession of k (not necessarily different) edges of the form
uv, vw, wx,..., yz.

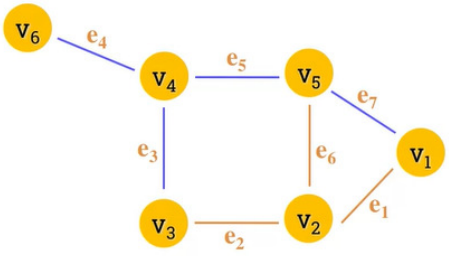
Example 1



v_1v_2, v_2v_3, v_3v_4 , v_4v_6 = e_1 ,e_2 ,e_3 ,e_4. = $v_1v_2v_3v_4v_6$

A walk of **length 4** from **v_1 to v_{6t}**

Example 2



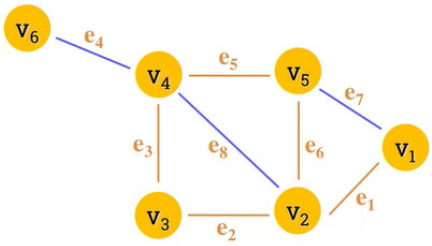
v_1v_2, v_2v_3, v_3v_2 , v_2v_5 = e_1 ,e_2 ,e_2 ,e_6. =. $v_1v_2v_3v_2v_5$

A walk of **length 4** from **v_1 to v_5** (passes twice through the edge e_2)

Trail

A **trail** is a walk in which no edge is repeated. In a trail, vertices can be repeated but no edge is ever repeated.

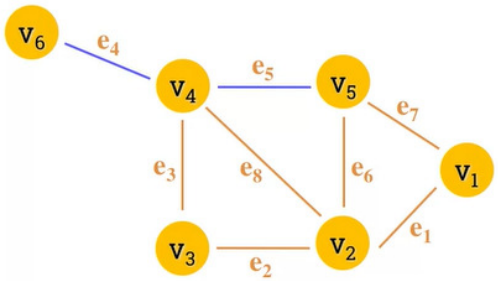
e1, e2, e3, e5, e6 is a trail



Circuit

A **circuit** is a is a closed trail. Circuits can have repeated vertices only.

e7, e6, e8, e3, e2, e1 is a circuit

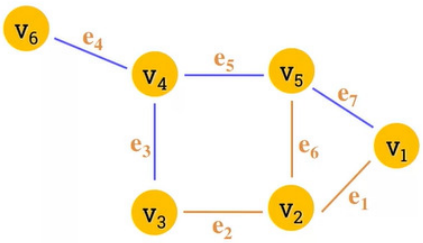


Definition of a path

A **path** is a trail in which neither vertices nor edges are repeated.

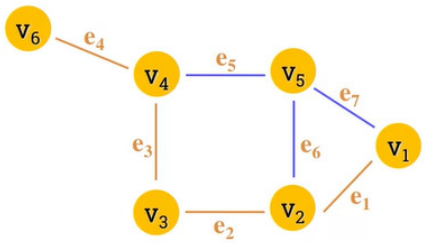
- The length of a path is given by the number of edges it contains

Example



$v_1v_2, v_2v_3, v_3v_2, v_2v_5 = e_1, e_2, e_2, e_6$. $v_1v_2v_3v_2v_5$

A walk of **length 4** from v_1 to v_5 but not a path

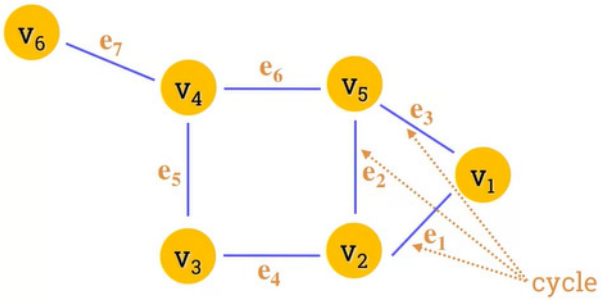


$v_1v_2, v_2v_3, v_3v_4, v_4v_6 = e_1, e_2, e_3, e_4$. $= v_1v_2v_3v_4v_6$

A path of **length 4** from v_1 to v_6

Cycle

A cycle is a closed path, consisting of edges and vertices where a vertex is reachable from itself.



$v_1v_2, v_2v_5, v_5v_1 = e_1, e_2, e_3 = v_1v_2v_5v_1$

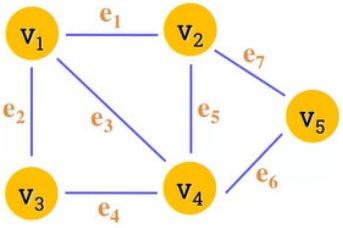
A walk of **length 3** from v_1 **to** v_1 = closed path = cycle

- Note** In example above for cycle, it is from V_1 to V_5, instead of V_1 to V_1.

Euler path

Definition: A **Eulerian path** in a graph is a path that uses each edge precisely once. If such a path exists, the graph is called **traversable**.

Example:



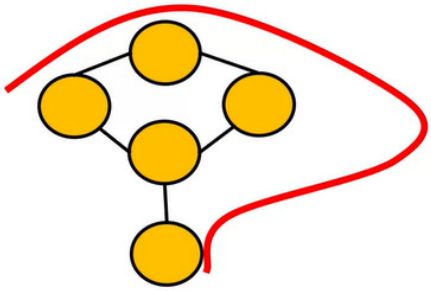
$e_2, e_4, e_6, e_7, e_5, e_3, e_1 = v_1v_3v_4v_5v_2v_4v_1v_2$

is a Euler path

Hamiltonian path

A **Hamiltonian path** (also called a *traceable path*) is a path that visits each vertex exactly once.

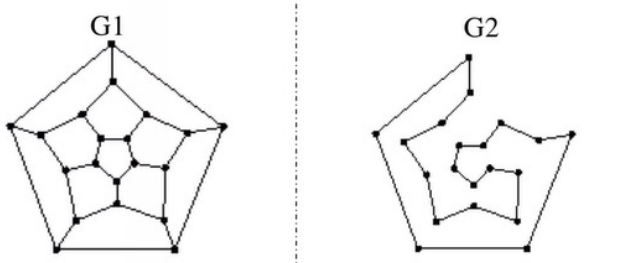
Example:



A graph that contains a Hamiltonian path is called a **traceable graph**.

Hamiltonian cycle

A **Hamiltonian cycle** is a cycle that visits each vertex exactly once (except for the starting vertex, which is visited once at the start and once again at the end).



Hamiltonian cycle

Hamiltonian graph

A graph that contains a Hamiltonian cycle is called a **Hamiltonian graph**.

Any Hamiltonian cycle can be converted to a Hamiltonian path by removing one of its edges.

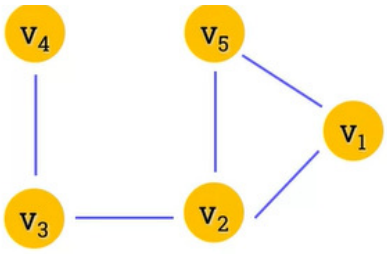
Connectivity

An ***undirected*** graph is ***connected*** if

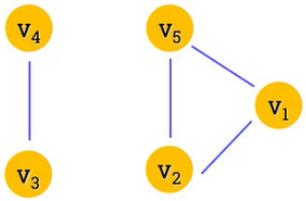
you can get from ***any node to any other*** by following a ***sequence of edges***

OR

any two nodes are **connected** by a path.



Connected graph

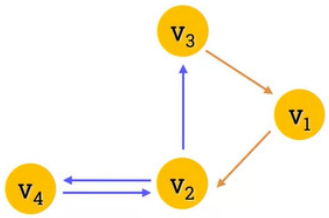


Not connected graph

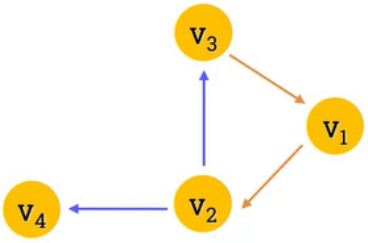
there is no path from (v_1, v_2 or v_6) to (v_3 or v_4)

Strong Connectivity

A directed graph is **strongly connected** if there is a **directed path** from any node to any other node.



Strongly connected directed graph

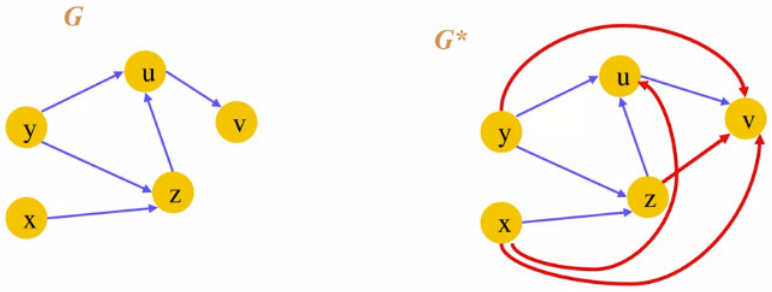


Not Strongly connected directed graph

No directed path from v_4 to any of the other 3 vertices

Transitive Closure

Given a digraph *G*, the transitive closure of *G* is the digraph *G** such that:
*G** has the same vertices as *G*
if *G* has a directed path from *u* to *v* ($u \neq v$), *G** has a directed edge from *u* to *v*



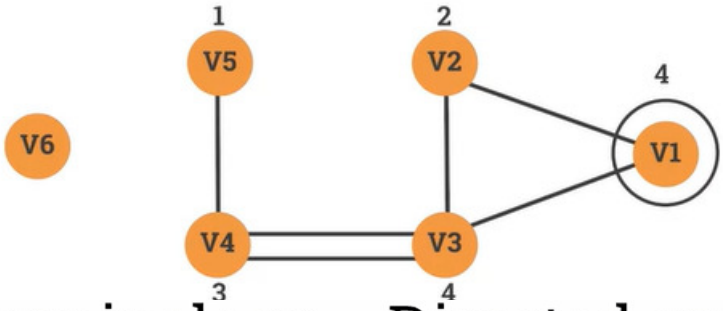
The transitive closure provides reachability information about a digraph.

Terminology – Undirected graphs

Degree of a vertex (deg (v)): the number of edges incident on v

A loop contributes **twice** to the degree

An isolated vertex has a degree : 0



Terminology – Directed graphs

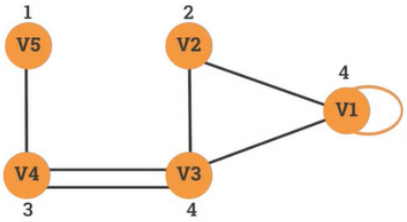
In-deg (v): number of edges for which v is the terminal vertex

Out-deg (v): number of edges for which v the is initial vertex

deg (v) = Out-deg (v) + IN-deg (v)

A loop contributes **twice** to the degree as it contributes 1 to both in-degree and out-degree.

Example



$\deg(v_1) = \text{in-deg}(v_1) + \text{out-deg}(v_1) = 2 + 2=4$
$\deg(v_2) = \text{in-deg}(v_2) + \text{out-deg}(v_2) = 1 + 1=2$
$\deg(v_3) = \text{in-deg}(v_3) + \text{out-deg}(v_3) = 2 + 2=4$
$\deg(v_4) = \text{in-deg}(v_4) + \text{out-deg}(v_4) = 1 + 2=3$
$\deg(v_5) = \text{in-deg}(v_5) + \text{out-deg}(v_5) = 1 + 0=1$
$\deg(v_6) = \text{in-deg}(v_6) + \text{out-deg}(v_6) = 0 + 0=0$

Degree sequence of a graph

Given an undirected graph G, a **degree sequence** is a **monotonic nonincreasing** sequence of the vertex degrees of all the vertices of G.

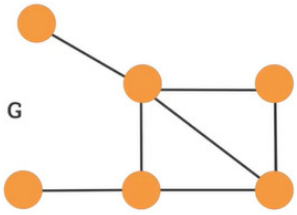
- Written in descending order separated by commas

Degree sequence property 1

The **sum of the degree sequence** of a graph is always **even**.

Therefore, it is impossible to construct a graph where the sum of the degree sequence is odd.

Example



The degree sequence of G is: 4,3,3,2,1,1

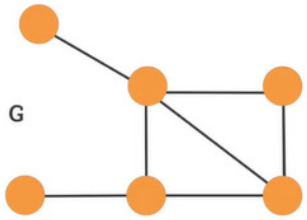
Sum of the degree sequence = 1+1+2+3+3+4 = 14

Degree sequence property 2

Given a graph G, the sum of the degree sequence of G is **twice** the number of edges in G.

Number of edges(G) = (sum of degree sequences of G) / 2

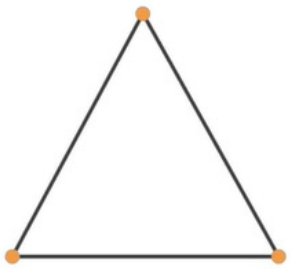
Example 1



The degree degree sequence of G is: 4,3,3,2,1,1

Number of edges = (1+1+2+3+3+4)/2 = 14/2 = 7

Example 2

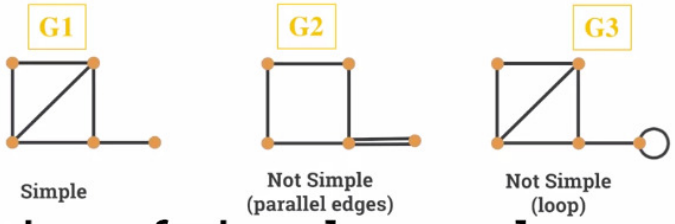


The degree sequence of G is: 2,2,2

Number of edges = (2+2+2)/2 = 6/2 = 3

Simple graphs

A *simple graph* is a graph without loops and parallel edges.



Properties of simple graphs

Given a simple graph G with n vertices, then the degree of each vertex of G is at most equal to n-1.

Proof :

Let v be a vertex of G such that deg(v) > n-1

However, we have only n-1 other vertices for v to be connected to

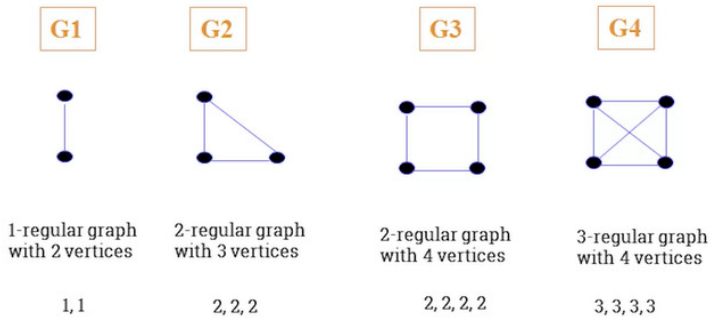
Hence, the other connections can only be a result of parallel edges of loops.

Regular graphs

A **graph** is said to be **regular** of degree if all local degrees are the same number.

A graph G where all the vertices the same degree, r, is called an r-regular graph.

Examples



Properties of regular graphs

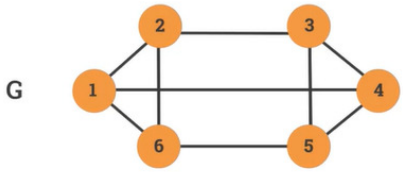
Given an r-regular G with n vertices, then the following is true:

Degree sequence of G = r,r,r,...,r (n times)

Sum of degree sequence of G = r x n

Number of edges in G = r x n/2

Example: 3-regular with 6 vertices

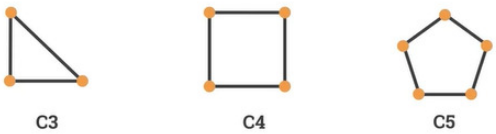


Degree Sequence = 3,3,3,3,3,3

Sum of degree sequence = 3x6=18

Number of edges = 18/2=9

Special regular graphs: cycles

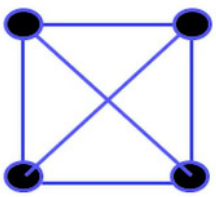


C_3 is 2-regular graph with 3 vertices
C_4 is 2-regular graph with 4 vertices
C_5 is 2-regular graph with 5 vertices

deg seq. of C_3 = 2,2,2
deg seq. of C_4 = 2,2,2,2
deg seq. of C_5 = 2,2,2,2,2

3-regular with 4 vertices

Sum of degree sequence = 3x4 = 12
The sum is even, hence it is possible to construct 3-regular graph with 4 vertices.



3-regular graph with 5 vertices

Sum of degree sequence = 3x5= 15

The sum if odd, hence it is impossible to construct a 3-regular graph with 5 vertices.

Complete graphs

A **complete graph** is a ***simple*** graph where ***every pair of vertices*** are ***adjacent*** (linked with an edge).

We represent a complete graph with n vertices using the symbol K_n.

Complete graph properties

A complete graph with n vertices, $\mathbf{k_n}$, has the following properties:

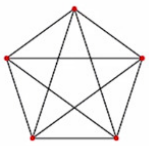
Every vertex has a degree **(n-1)**

Sum of degree sequence = **n(n-1)**

Number of edges = **n(n-1)/2**

Another example of a complete graph

k_5



There are 5 vertices
Degree of each vertex = (5-1) = 4
Sum of deg. Seq. = 5(5-1) = 20
Number of edges = 5(5-1)/2=20/2=10

Summary

In this week, we learned what graph is, how it is defined by its edges, vertices & direction. Alongside this, we explored the different configuration of vertices & edges that result in a path, circuit, cycle and etc. Also we looked at the meaning of a degree sequences and how to count the degree of each vertex with in/out degree for vertices. Finally, we examined special graphs like simple, r-regular and complete graphs.