Discrete Structures

Graph Theory

Nathan Warner

Computer Science
Northern Illinois University
August 31, 2023
United States

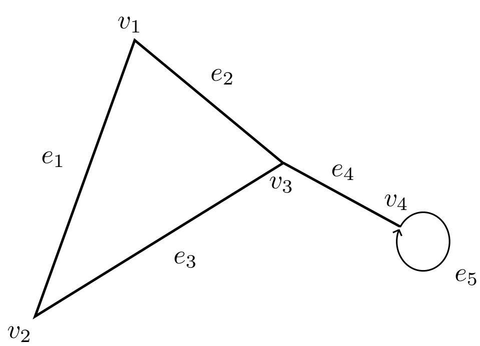
Contents

1 Graphs

Definition 1:

A graph G consists of two finite sets: a nonempty set $V(G)$ of vertices and a set $E(G)$ of edges, where each edge is associated with a set consisting of either one or two vertices called its endpoints. Formally, a graph is defined as an ordered pair $G = (V, E)$, where V is the set of vertices and E is the set of edges

$$G = (V, E)$$
$$V = \{v_1, v_2, v_3, ..., v_n\}$$
$$E = \{e_1, e_2, e_3, ..., e_m\}.$$



$$V = \{v_1, v_2, v_3, v_4\}$$
$$E = \{e_1, e_2, e_3, e_4, e_5\}.$$

We can also represent the edges by only stating the vertices which connect the edges

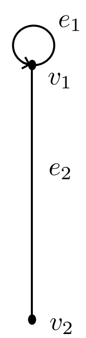
Edges	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_1, v_3\}$
e_3	$\{v_2, v_3\}$
e_4	$\{v_3, v_4\}$
e_5	$\{v_4\}$

2 Subgraphs

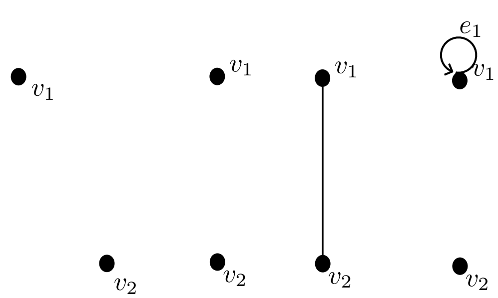
Definition 2:

Graph H is said to be a subgraph of a graph H iff every vertex in H is also a vertex in G, every edge in H is also an edge in G, and every edge in H has the same endpoints as it has in G.

Consider the graph:



Then the possible **sub graphs** could be:



Note:-

These graphs are not **all** the possibilites, just a few.

3 Degree

Definition 3:

In graph theory, the **degree** of a vertex refers to the number of edges that are connected to that vertex.

Definition 4:

Parallel edges are two or more edges that have the same pair of end vertices.

Definition 5:

Multiple Edges is a term used interchangeably with parallel edges.

Definition 6:

An **isolated vertex** is a vertex that has a degree of zero

Definition 7:

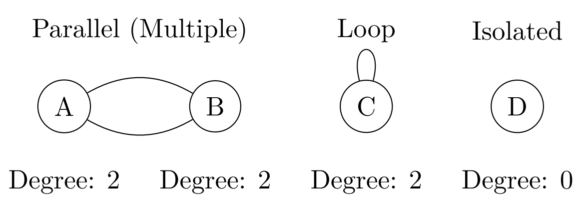
A **loop** is an edge that connects a vertex to itself.

Definition 8:

A **Degree Sequence** is an **n-tuple** of the degrees on vertices, in increasing order and with repetition.

Definition 9:

The **overall degree** is the sum of all the degrees.



4 Sum of Degrees and Vertices Theorem

Definition 10:

To denote the number of vertices in a graph, we say $||V||$, or just $|v|$. To denote the number of edges in a graph, we say $||E||$, or just $|E|$

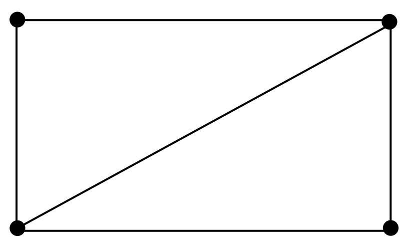
Definition 11:

The number of vertices in a graph is called the **order** of the graph

Definition 12:

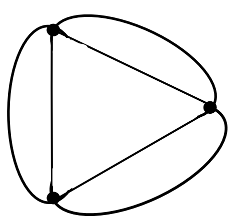
The number of edges in a graph is called the **size** of the graph.

Consider the graphs:



Then we have:

$$||V|| = 4$$
$$||E|| = 5$$
$$\sum deg = 10.$$



$$||V|| = 3$$
$$||E|| = 6$$
$$\sum deg = 12.$$

So you might notice from these two examples that the total degree of the graph ($\sum deg$) is exactly **twice** the number of edges. Thus, we can conclude:

Theorem 1

$$\sum deg = 2||E||.$$

Proof. Let G be a graph, that has n vertices $v_1, v_2, v_3, v_4, ..., v_n$ and m edges, where n is a positive integer and m is a nonnegative integer.

If e_1 is an edge, then

$$v_i, v_j = \begin{cases} 1 \ edge, \ 1V & \rightarrow degree = 2 \\ 1 \ edge, \ 2V & \rightarrow degree = 2 \end{cases} \tag{1}$$

Thus, no matter the case, the edge always contributes 2 to the total degree. ☺

Corollary 1. The total degree of a graph is even.

Corollary 2. In any graph, there are an even number of vertices of odd degree.

5 Adjacency and Incidence

Definition 13:

vertices that are connected by an edge are adjacent

Definition 14:

A vertex with a loop is adjacent to itself

Definition 15:

Two edges that share a vertex are adjacent

Definition 16:

An edge is incident on its endpoints

Definition 17:

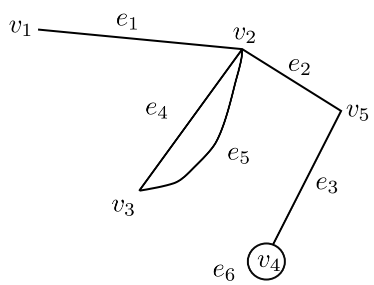
A vertex on which no edges are incident is an isolated vertex.

6 Adjacency Matrix

Definition 18:

Let G be a graph with vertices labeled $\{1, 2, 3, ..., n\}$. Then the **Adjacency Matrix** of G is the $n \times n$ matrix whose ij^{th} term is the number of the edges joining vertex i and vertex j

Consider the graph:



Since we have 5 vertices, then we will have a 5×5 matrix. Thus, our matrix for this graph will be:

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

To make things clearer, here is how the rows and columns are labeled:

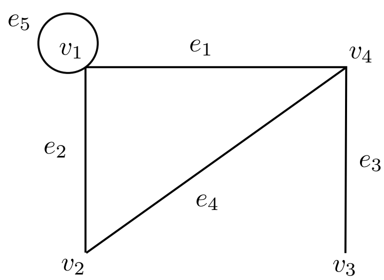
	v_1	v_2	v_3	v_4	v_5
v_1	0	1	0	0	0
v_2	1	0	2	0	1
v_3	0	2	0	0	0
v_4	0	0	0	1	1
v_5	0	1	0	1	0

7 Incidence Matrix

Definition 19:

An **incidence matrix** is a rectangular matrix B where $B[i][j]$ represents the relationship between vertex i and edge j.

Suppose we have the graph:



Then we write the **incidence matrix** as follows:

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Where the vertices as labeled vertically, and the edges are labeled horizontally, as such:

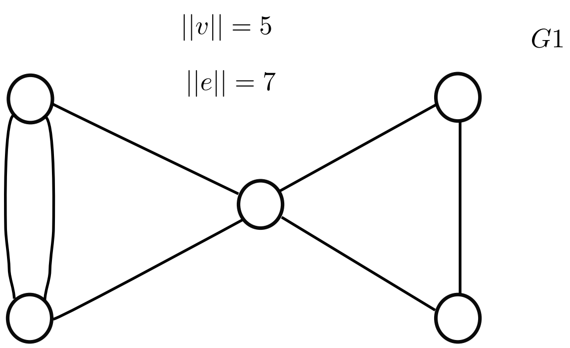
	e_1	e_2	e_3	e_4	e_5
v_1	1	1	0	0	2
v_2	0	1	0	1	0
v_3	0	0	1	0	0
v_4	1	0	1	1	0

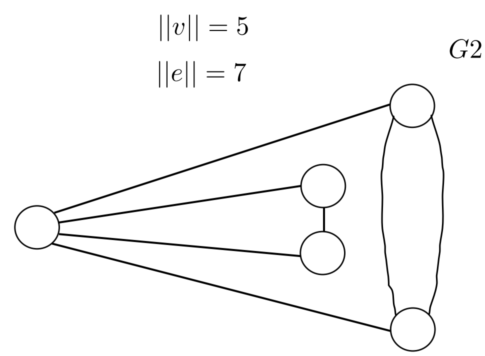
8 Isomorphism

Definition 20:

Two graphs G1 and G2 are isomorphic if they have the same number of vertices, edges, and there exists a matching between their vertices so that two vertices are connected by an edge in G1 if and only if corresponding vertices are connected by an edge in G2.

Consider the graphs:





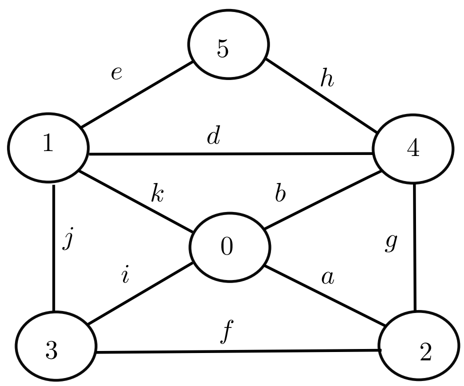
We can then see that these two graphs are **isomorphic**

9 Walks, Trails, Paths, and Circuits

Definition 21:

For the graph G, and vertices V and W, a **walk** from V to W is a finite alternating sequence of adjacent vertices and edges of G
The **length** of a walk is the number of edges in the walk.
A **trivial walk** is a walk with length zero.
A **closed walk** is a walk that starts and ends at the same vertex
An **open walk** is a walk that starts and ends at different vertices

Suppose we have the graph:



Then we can say a possible walk from 1 to 2 could be:

$$W = 1\ e\ 5\ h\ 4\ g\ 2.$$

Definition 22:

A **Trail** from v to w is a walk from v to w that does not contain a repeated edge.

Definition 23:

A **Path** from v to w is a trail that does not contain a repeated vertex. So, by inheritance, a path can also have **no** repeated edges.
The **distance** between two vertices is the length of the shortest path between those two vertices

$$d(v_1, v_2).$$

Definition 24:

A **Circuit** is a trail that contains at least one edge and starts and ends at the same vertex

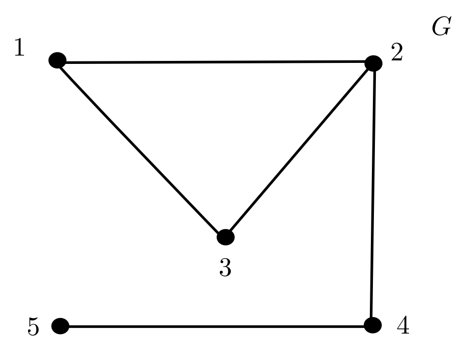
10 Eccentricity, Diameter, and Radius

Definition 25:

The **Eccentricity** of a vertex is the distance from v to a vertex farthest from v

$$\begin{aligned} &ecc(v) \\ or: \ &e(v). \end{aligned}$$

Consider the graph:



Thus we have:

$$\begin{aligned} d(1,2) &= 1 \\ d(1,3) &= 1 \\ d(1,4) &= 2 \\ d(1,5) &= 3. \end{aligned}$$

From these observations, we can deduce:

$$\begin{aligned} ecc(1) &= 3 \\ ecc(2) &= 2 \\ ecc(3) &= 3 \\ ecc(4) &= 2 \\ ecc(5) &= 3. \end{aligned}$$

Definition 26:

The **diameter** of a graph G is the maximum vertex eccentricity.
The **radius** of a graph G is the minimum vertex eccentricity.
If $ecc(v) = diam(G)$, then v is a **peripheral vertex**
If $ecc(v) = rad(G)$, then v is a **central vertex**

11 Connectedness

Definition 27:

- A **graph is connected** iff there is a walk between each pair of vertices
- A **disconnecting set** for a graph G is a set of edges whose removal disconnects G
- A **cut set** is a disconnecting set such that no proper subset of the disconnecting set is disconnecting
- A **bridge** is a disconnecting set that has a cardinality of 1
- **Edge connectivity** represents the minimum number of edges that you have to remove such that you get the graph to be disconnected

  $$\text{Edge connectivity: } \lambda(G).$$

- A **separating set** is a set of vertices whos removal will cause a disconnection in the graph.

  Note: Deletion of a vertex in a graph will also remove any edges that are connected to that vertex.
- A **cut-vertex** is a vertex whose removal causes the graph to be disconnected and split into components
- **Vertex connectivity** is the minimum number of vertices that must be removed to cause a disconnection.

  $$\text{Vertex connectivity: } \kappa(G).$$

12 Euler Trails and Circuits

Definition 28:

- An **Euler trail** is a trail that visits every edge exactly once.
- **If all vertices have even degrees except two, we can declare that the graph has a Euler trail**
- An **Euler circut** is an Euler trail that visits every edge exactly once and starts and ends at the same vertex.
- **If all vertices have even degrees then we have an Euler circuit**
- A connected graph G is **Eulerian** iff the degree of each vertex is even

13 Fleury's Algorithm

Definition 29:

Fleury's algorithm is an algorithm that is used to find Euler trails and circuits

Steps (Euler Trail):

1. Start at odd vertex
2. Verify that the requirements for an euler trail or circuit are satisfied
3. Make a replica of the graph
4. Pick an edge, then delete it from the replica, if the deletion of the edge causes would cause the graph to become disconnected, do not remove the edge.

Steps (Euler Circuit):

1. Start at any vertex
2. Verify that the requirements for an euler trail or circuit are satisfied
3. Make a replica of the graph
4. Pick an edge, then delete it from the replica, if the deletion of the edge causes would cause the graph to become disconnected, do not remove the edge.

14 Hamiltonian paths and circuits

Definition 30:

- A **Hamiltonian path** is a path that visits every vertex exactly once
- A **Hamiltonian circuit** is a circuit that contains each vertex in G exactly once, except for the starting and ending vertex that appears twice.
- A **Hamiltonian graph** is a graph that has a Hamiltonian circuit

15 Ore's Theorem

Theorem 2

If a simple graph with $n \geqslant 3$ vertices, and if deg(v) + deg(w) $\geqslant n$ for each pair of non-adjacent vertices v and w, then G is Hamiltonian

16 The shortest path problem

Definition 31:

A **weighted graph** is a graph whose edges have some weight (value)

16.1 Dijkstran's algorithm

1. Pick starting and ending vertices
2. Give a value of zero to your starting vertex, and give the other vertices a value of infinity
3. Pick a vertex to travel to from your starting point
4. Add the weight of the edge to the vertex
5. If the value is less than the vertex you are traveling to, update it.
6. Once you travel to all possible vertices that are connected to your starting vertex, you are done with that vertex
7. Repeat with other vertices