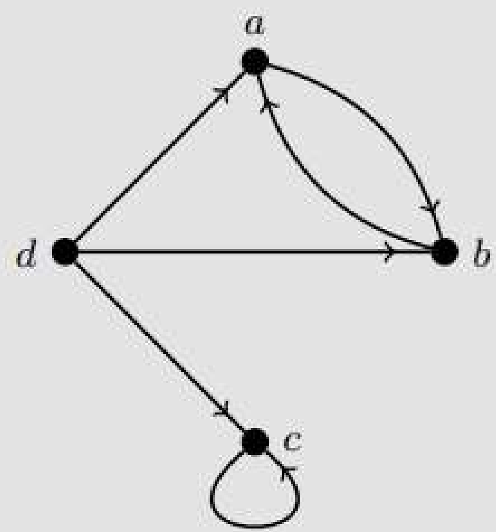
National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

LECTURE # 03

Definition 1.6 A ***directed graph***, or ***digraph***, is a graph $G = (V, A)$ that consists of a vertex set $V(G)$ and an *arc set* $A(G)$. An ***arc*** is an ordered pair of vertices.

Example 1.6 Let G_5 be a digraph where $V(G_5) = \{a, b, c, d\}$ and $A(G_5) = \{ab, ba, cc, dc, db, da\}$. A drawing of G_5 is given below.



Path Graphs

The Path Graph P_n, $n \geq 2$, consists of n vertices $v_1, \ldots, v_n$ and $n-1$ edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

Path Graphs

The Path Graph P_n, $n \geq 2$, consists of n vertices $v_1, \ldots, v_n$ and $n-1$ edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

The Graph P_5

Path Graphs

The Path Graph P_n, $n \geq 2$, consists of n vertices $v_1, \ldots, v_n$ and $n - 1$ edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

The Graph P_2

Path Graphs

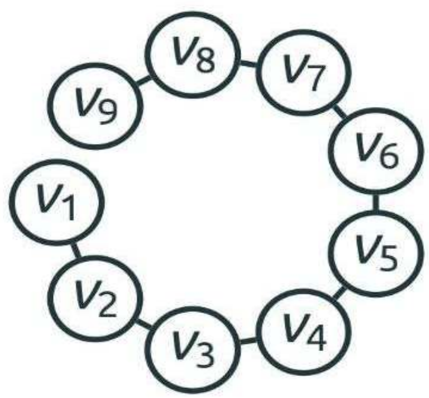
The Path Graph P_n, $n \geq 2$, consists of n vertices $v_1, \ldots, v_n$ and $n-1$ edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

The Graph P_9

Path Graphs

The Path Graph P_n, $n \geq 2$, consists of n vertices $v_1, \ldots, v_n$ and $n-1$ edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$

The Graph P_9



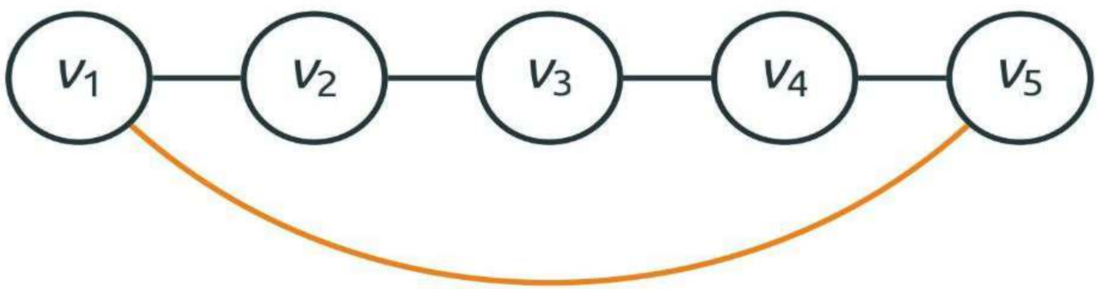
Cycle Graphs

The Cycle Graph C_n, $n \geq 3$, consists of n vertices $v_1, \ldots, v_n$ and n edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

Cycle Graphs

The Cycle Graph C_n, $n \geq 3$, consists of n vertices $v_1, \ldots, v_n$ and n edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

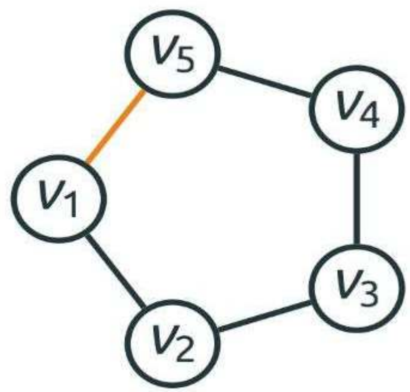
The Graph C_5



Cycle Graphs

The Cycle Graph C_n, $n \geq 3$, consists of n vertices $v_1, \ldots, v_n$ and n edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

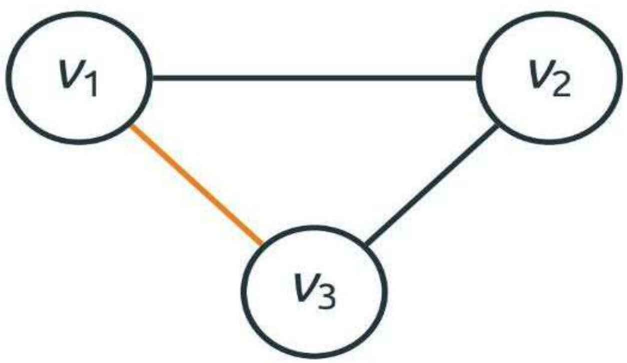
The Graph C_5



Cycle Graphs

The Cycle Graph C_n, $n \geq 3$, consists of n vertices $v_1, \ldots, v_n$ and n edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

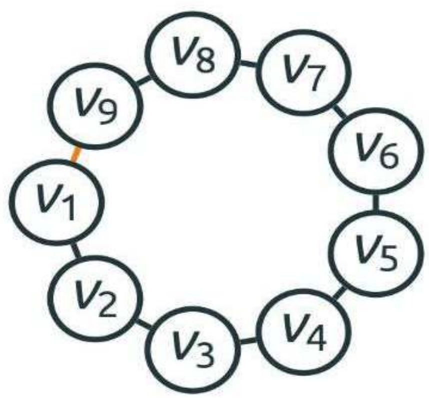
The Graph C_3



Cycle Graphs

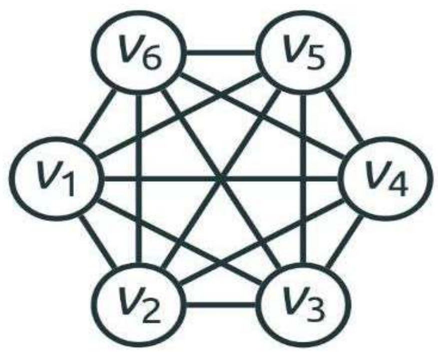
The Cycle Graph C_n, $n \geq 3$, consists of n vertices $v_1, \ldots, v_n$ and n edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

The Graph C_9



Complete Graphs

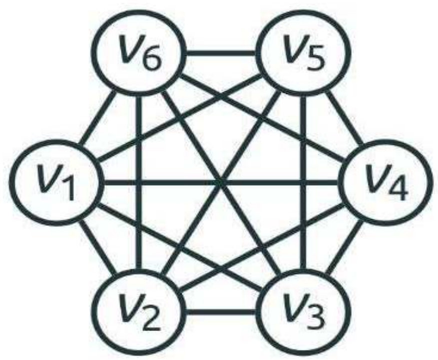
The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)



Complete Graphs

The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

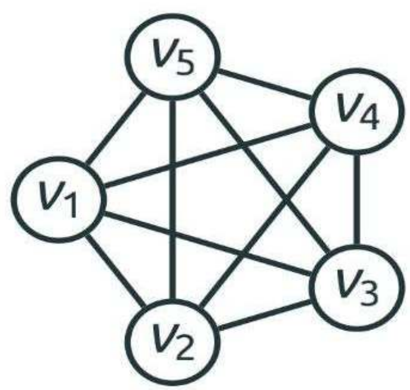
The Graph K_6



Complete Graphs

The **Complete Graph (Clique)** K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

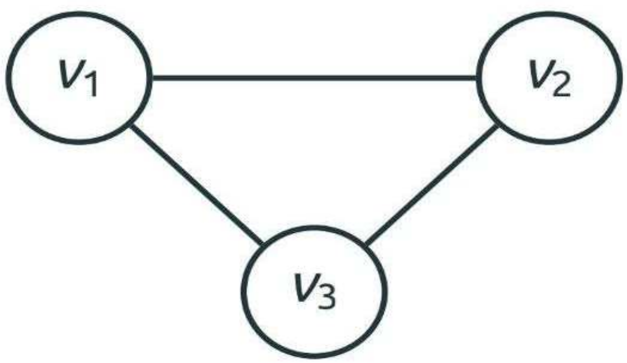
The Graph K_5



Complete Graphs

The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

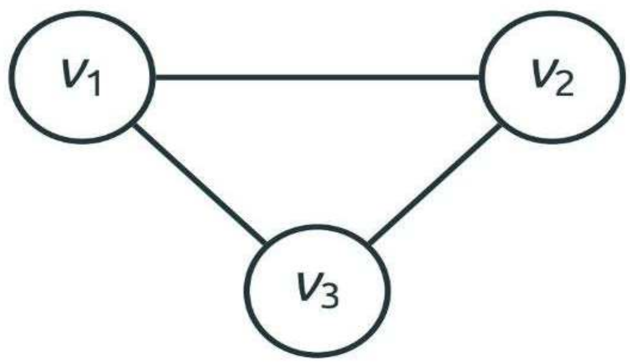
The Graph K_3



Complete Graphs

The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

The Graph $K_3 = C_3$



Complete Graphs

The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

The Graph K_2

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology FAST

Complete Graphs

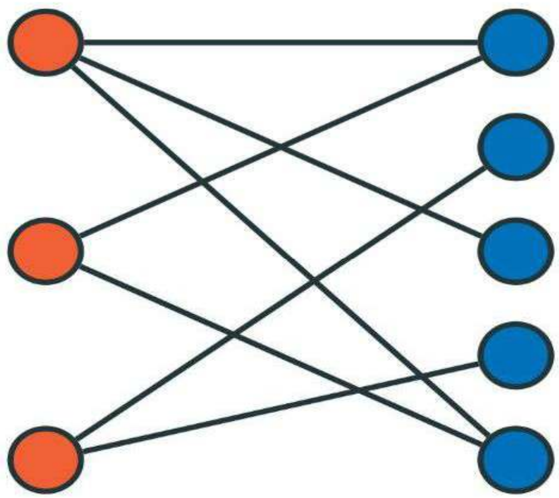
The Complete Graph (Clique) K_n, $n \geq 2$, contains n vertices $v_1, \ldots, v_n$ and all edges between them ($n(n-1)/2$ edges)

The Graph $K_2 = P_2$

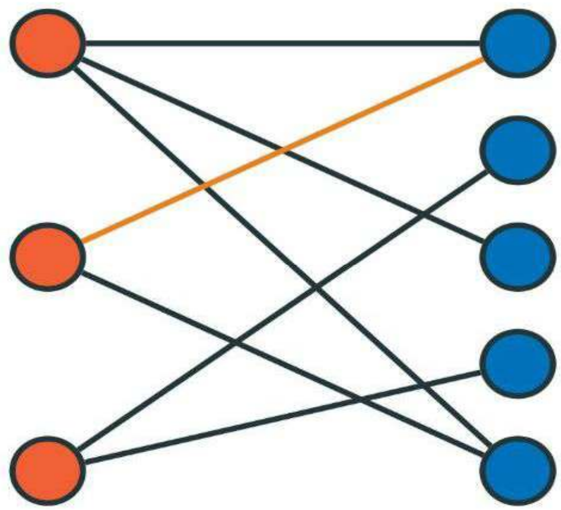
Bipartite Graphs

- A graph G is Bipartite if its vertices can be partitioned into two disjoint sets L and R such that
  - Every edge of G connects a vertex in L to a vertex in R
  - I.e., no edge connects two vertices from the same part
- L and R are called the parts of G

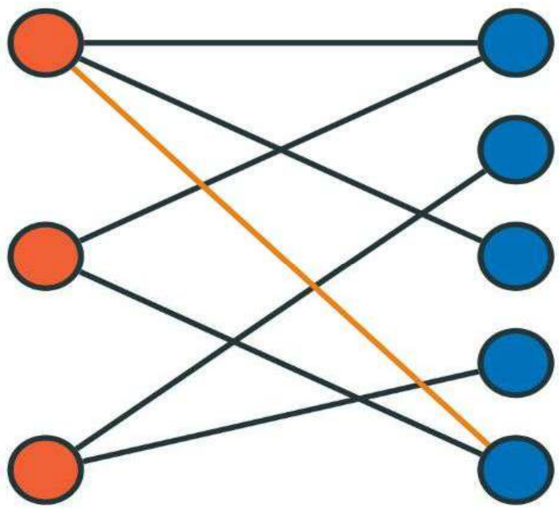
Bipartite Graphs: Examples



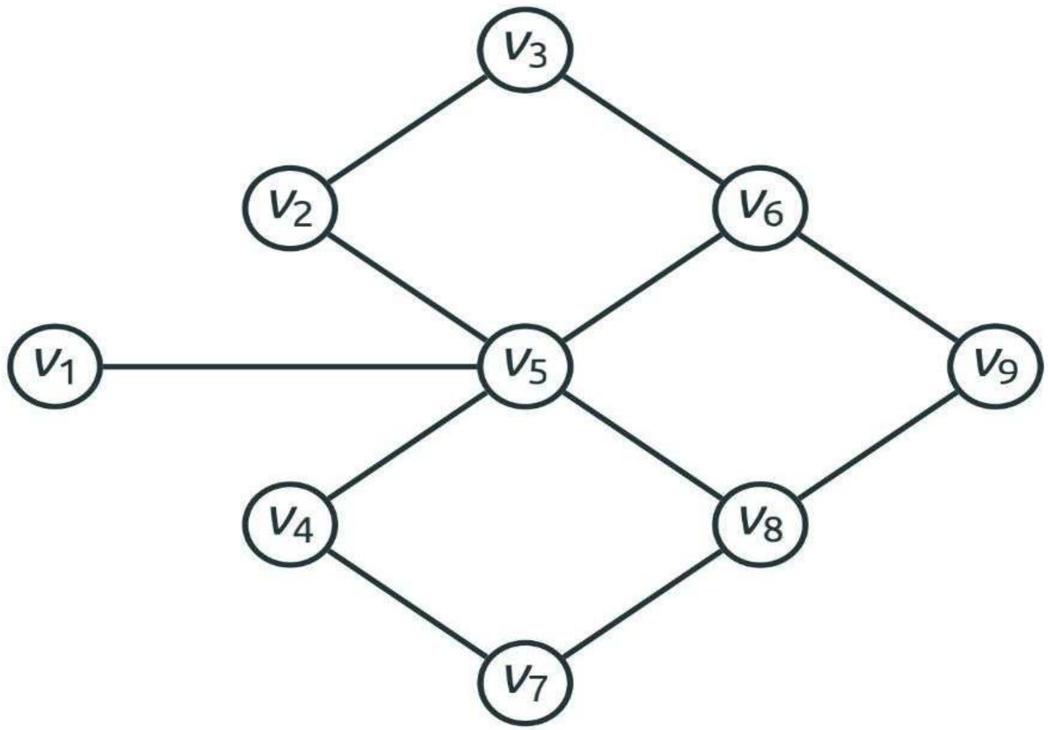
Bipartite Graphs: Examples



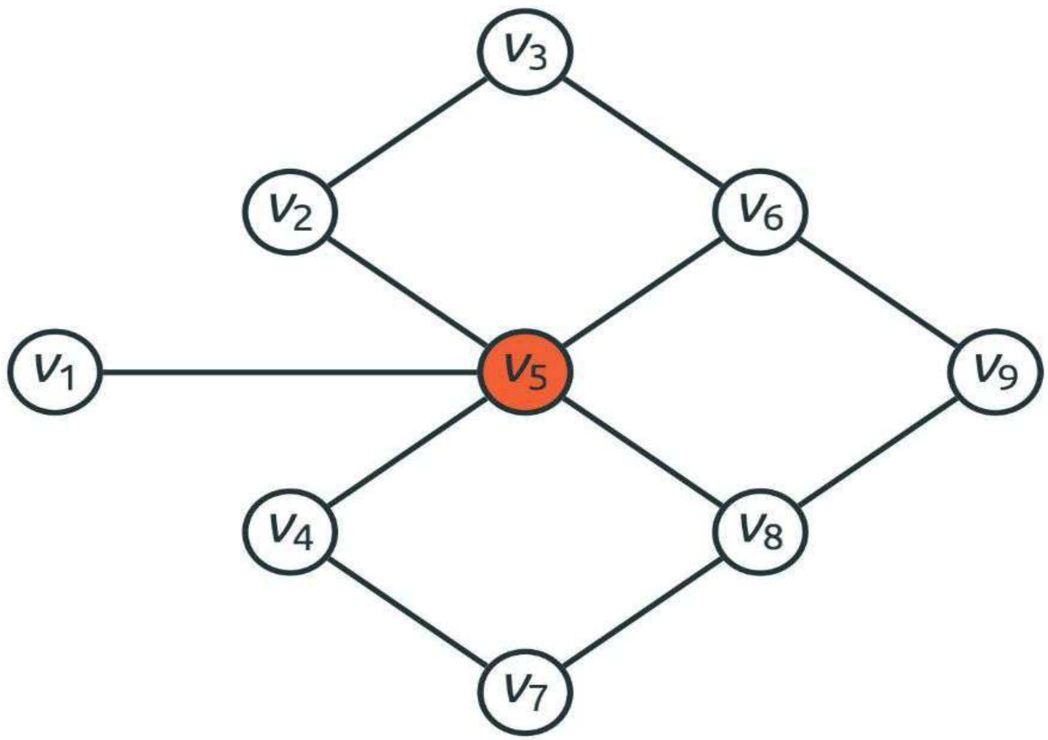
Bipartite Graphs: Examples



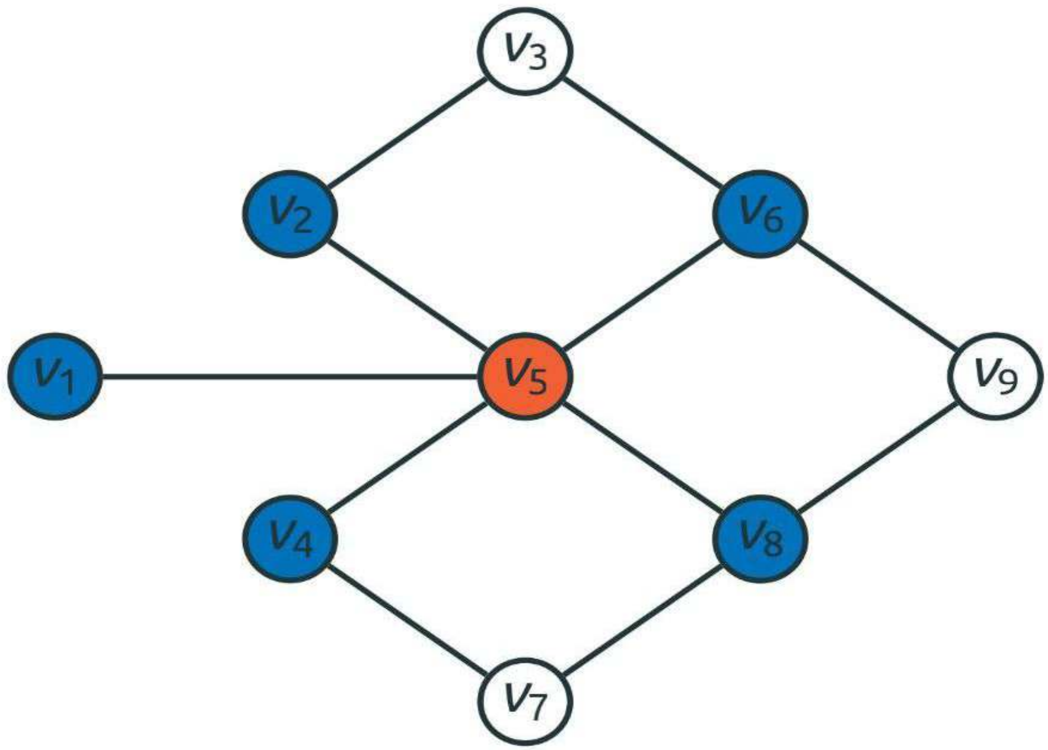
Bipartite Graphs: Examples



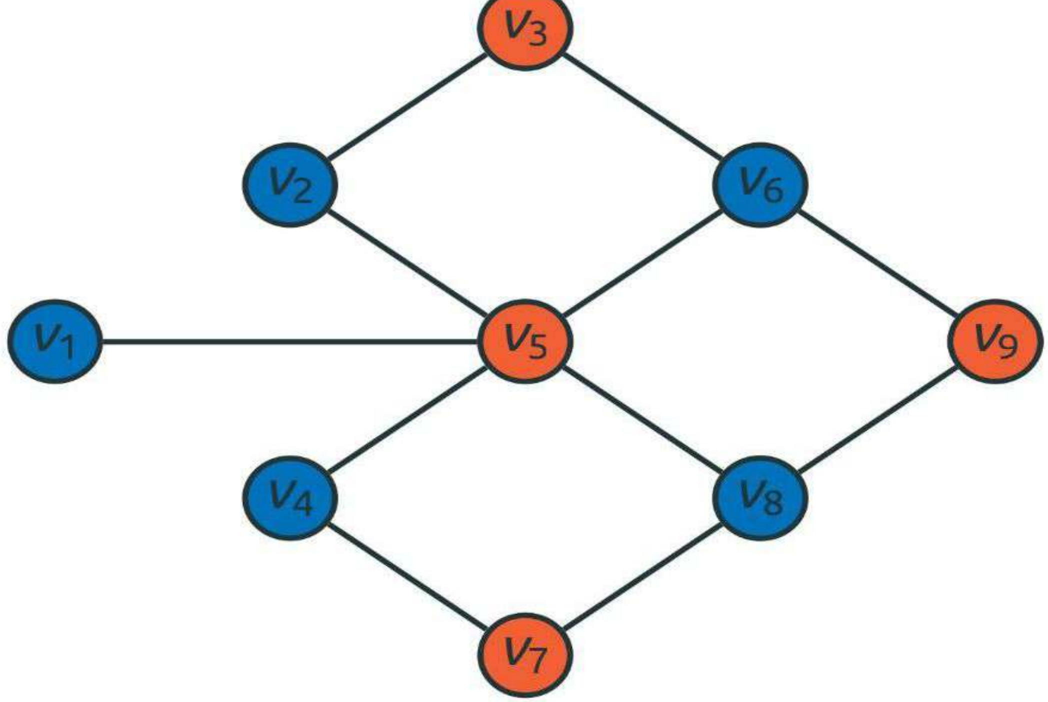
Bipartite Graphs: Examples



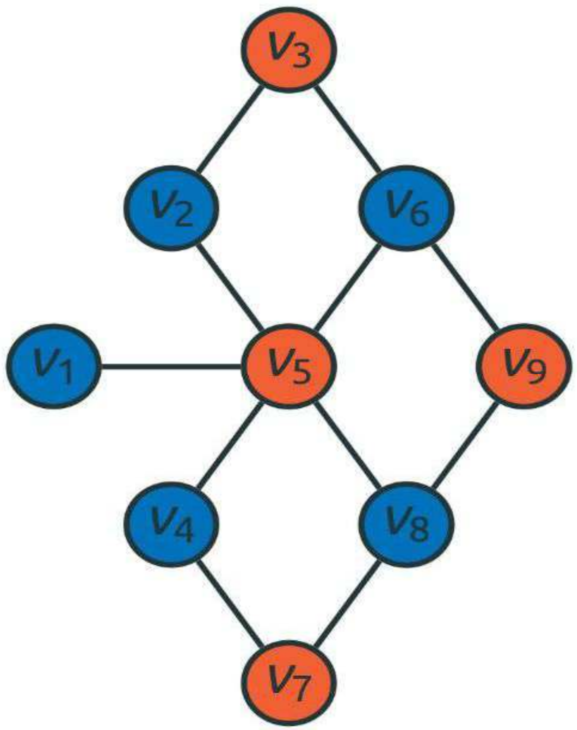
Bipartite Graphs: Examples



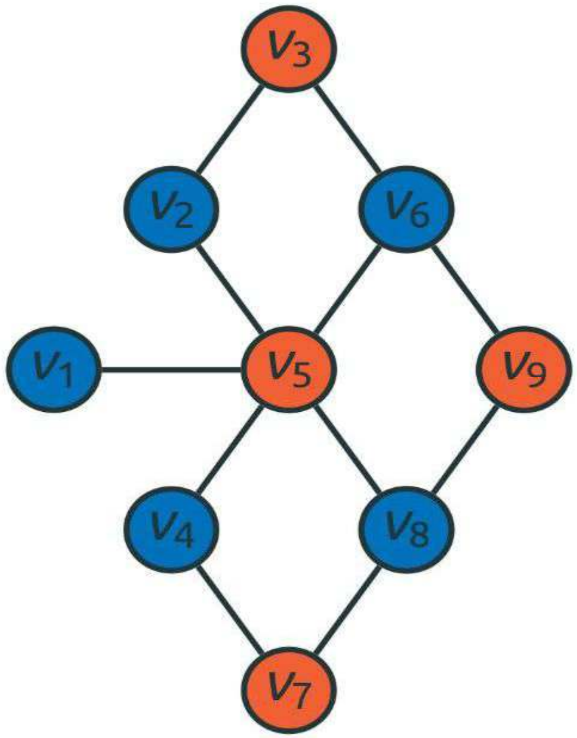
Bipartite Graphs: Examples

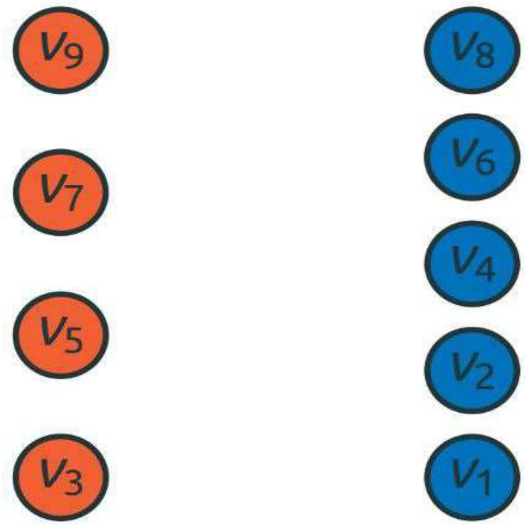


Bipartite Graphs: Examples

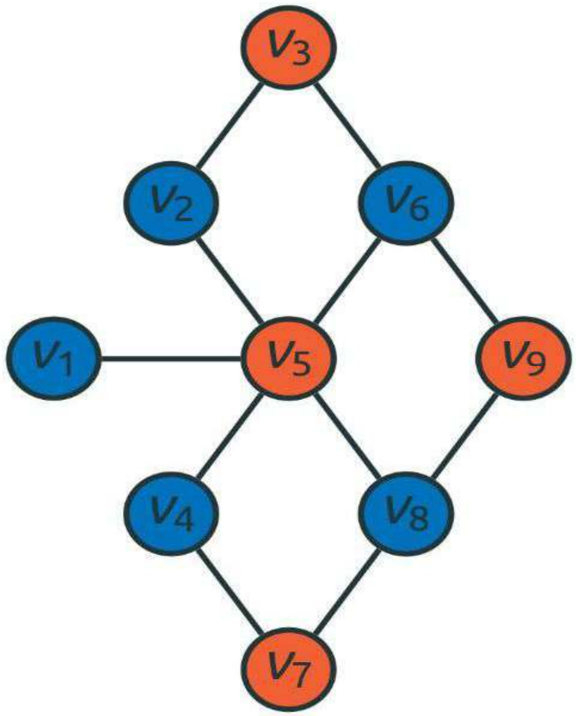


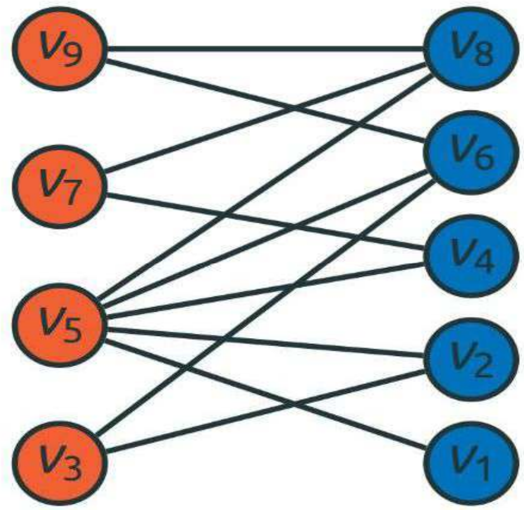
Bipartite Graphs: Examples





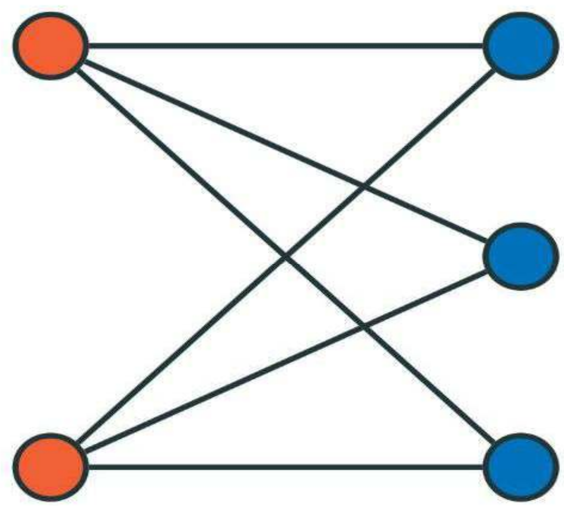
Bipartite Graphs: Examples





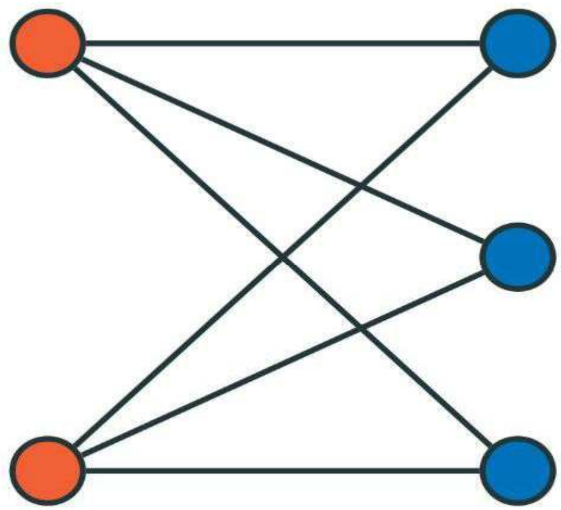
Complete Bipartite Graphs

Complete bipartite graph

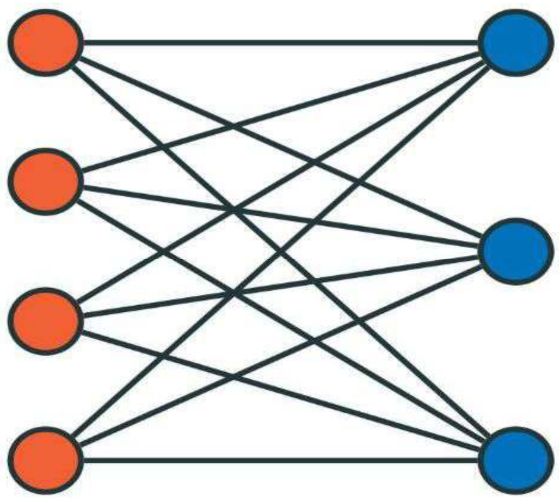


Complete Bipartite Graphs

Complete bipartite graph $K_{2,3}$

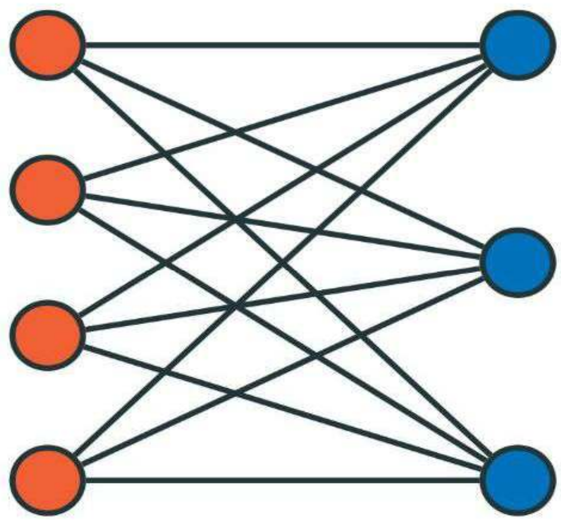


Complete Bipartite Graphs



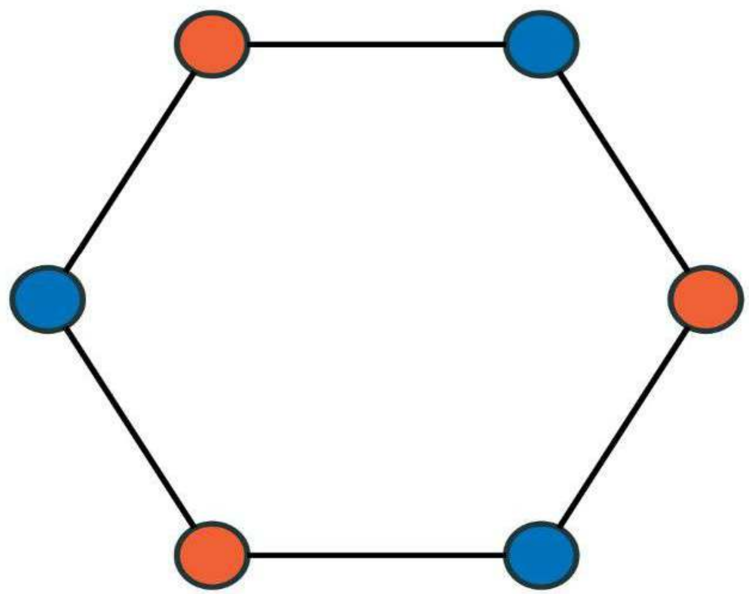
Complete Bipartite Graphs

Complete bipartite graph $K_{4,3}$



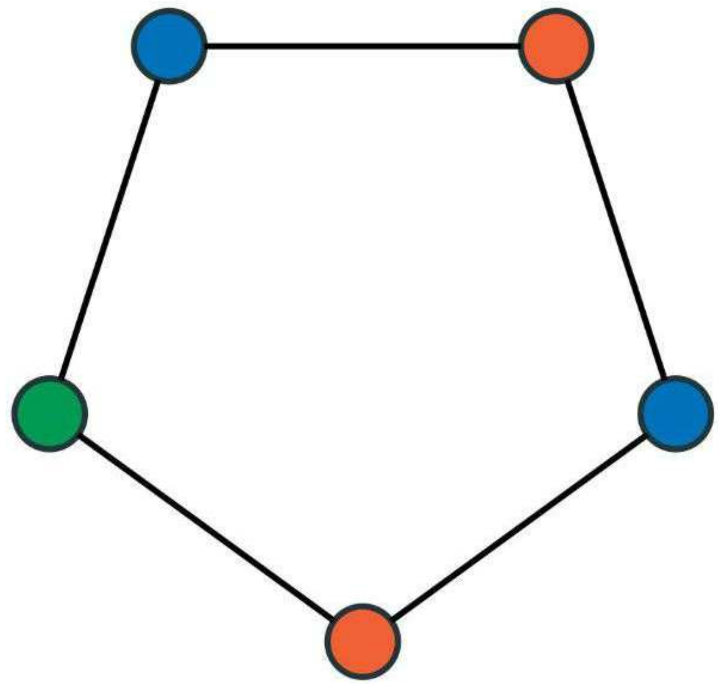
Cycle Graphs

For even n, C_n is bipartite



Cycle Graphs

For odd $n > 2$, C_n is not bipartite



National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

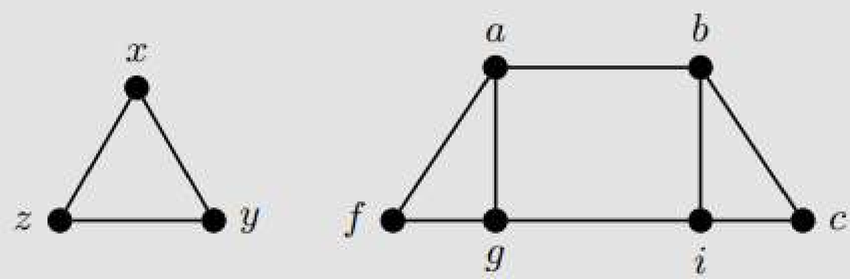
Graph Combinations

Definition 1.15 Given two graphs G and $\cancel{G}$ H the ***union*** $G \cup H$ is the graph with vertex-set $V(G) \cup V(H)$ and edge-set $E(G) \cup E(H)$.

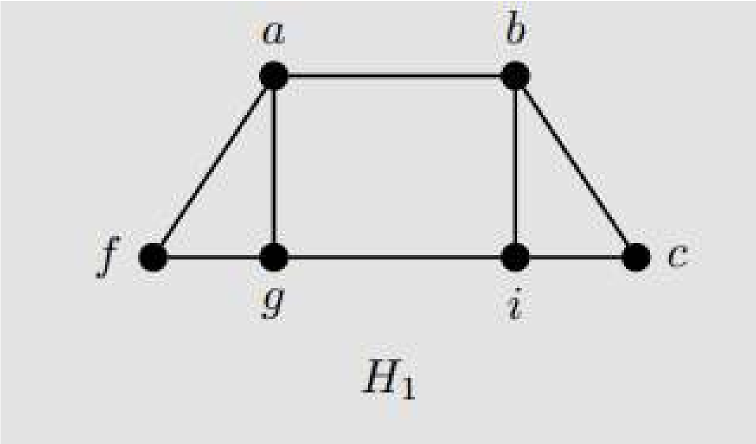
If the vertex-sets are disjoint (that is $V(G) \cap V(H) = \emptyset$) then we call the disjoint union the ***sum***, denoted $G + H$.

Example 1.12 Find the sum $K_3 + H_1$ and the union $H_1 \cup H_4$ using the graphs from Examples 1.3 and 1.4.

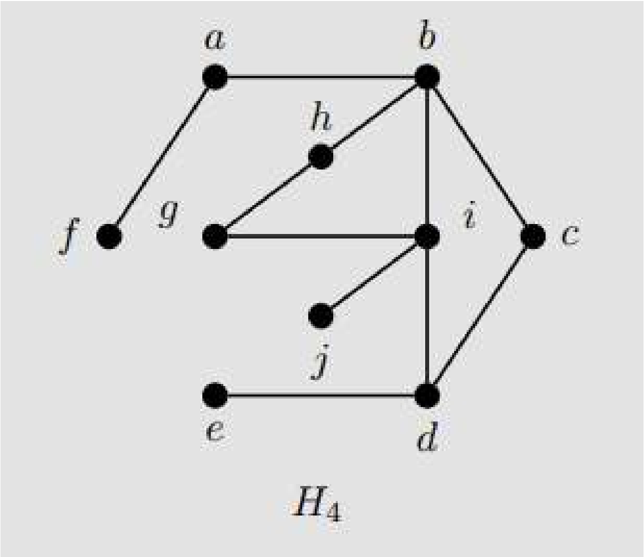
Solution: First note that, since we are finding the sum $K_3 + H_1$, we are assuming the vertex sets are disjoint. Thus the resulting graph is simply the graph below.

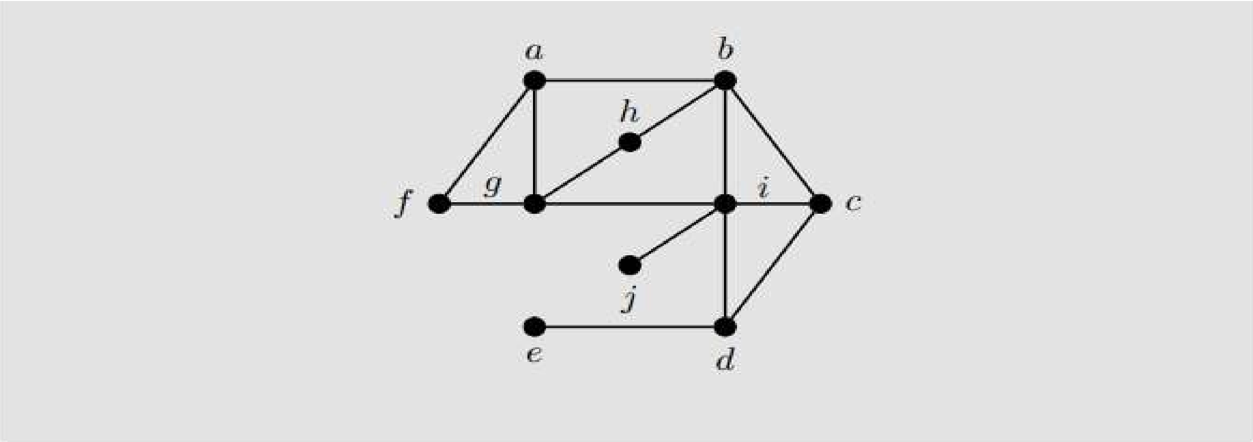


Next, since H_1 and H_4 are subgraphs of the same graph and have some edges in common, their union will consist of all the edges in at least one of H_1 and H_4, where we do not draw (or list) an edge twice if it appears in both graphs, as shown in the following graph.



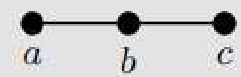
$\cup$



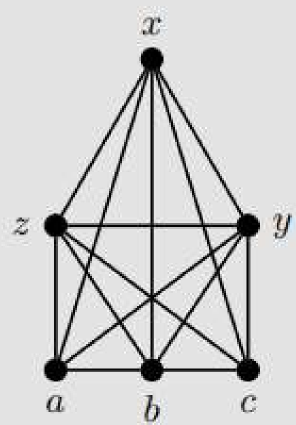


Definition 1.16 The ***join*** of two graphs G and H, denoted $G \vee H$, is the sum $G + H$ together with all edges of the form xy where $x \in V(G)$ and $y \in V(H)$.

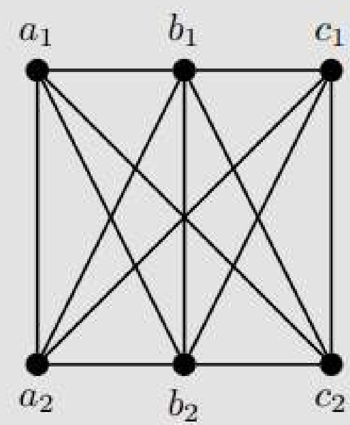
Example 1.13 Find the join of K_3 and the graph G below consisting of three vertices and two edges, as well as the join $G \vee G$.



Solution: The join $K_3 \vee G$ is shown below on the left. Note that every vertex from K_3 is adjacent to all those from G, but this is not K_6 since the edge ac is missing. The join $G \vee G$ is on the right below.



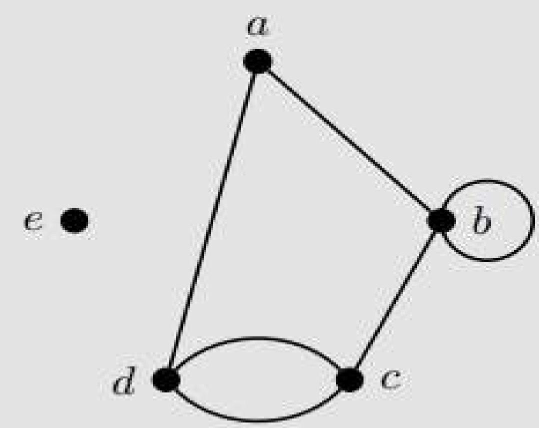
$K_3 \vee G$



$G \vee G$

Quick Review

Example 1.1 Let G_4 be a graph where $V(G_4) = \{a, b, c, d, e\}$ and $E(G_4) = \{ab, cd, cd, bb, ad, bc\}$. Although G_4 is defined by these two sets, we generally use a visualization of the graph where a dot represents a vertex and an edge is a line connecting the two dots (vertices). A drawing of G_4 is given below.

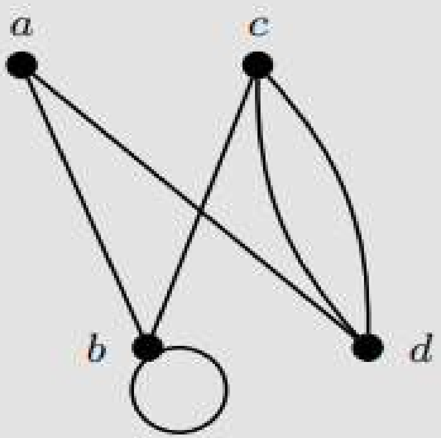


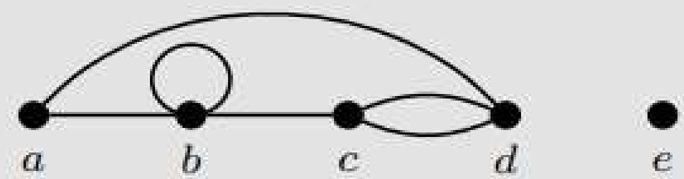
Note that two lines were drawn between vertices c and d as the edge cd is listed twice in the edge set. In addition, a circle was drawn at b to indicate an edge (bb) that starts and ends at the same vertex.

Definition 1.2 The number of vertices in a graph G is denoted $|V(G)|$, or more simply $|G|$. The number of edges is denoted $|E(G)|$ or $\|G\|$.

Definition 1.4 A ***subgraph*** H of a graph G is a graph where H contains some of the edges and vertices of G; that is, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

Example 1.2 Consider the graph G_4 from Example 1.1. Below are two different drawings of G_4.





To verify that these drawings represent the same graph from Example 1.1, we should check the relationships arising from the vertex set and edge set. For example, there are two edges between vertices c and d, a loop at b, and no edges at e. You should verify the remaining edges.

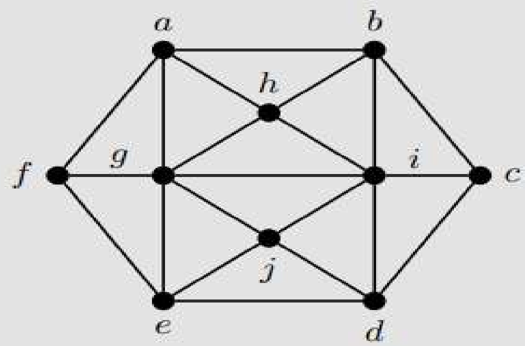
Definition 1.3 Let G be a graph.

- If xy is an edge, then x and y are the ***endpoints*** for that edge. We say x is ***incident to*** edge e if x is an endpoint of e.
- If two vertices are incident to the same edge, we say the vertices are ***adjacent***, denoted $x \sim y$. Similarly, if two edges share an endpoint, we say they are adjacent. If two vertices are adjacent, we say they are ***neighbors*** and the set of all neighbors of a vertex x is denoted $N(x)$.
  - ab and ad are adjacent edges in G_4 since they share an endpoint, namely vertex a

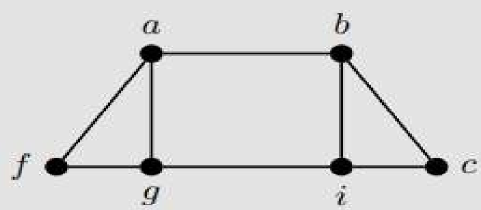
- $a \sim b$, that is a and b are adjacent vertices as ab is an edge of G_4
- $N(d) = \{a, c\}$ and $N(b) = \{a, b, c\}$

- If two vertices (or edges) are not adjacent then we call them ***independent***.
- If a vertex is not incident to any edge, we call it an ***isolated vertex***.
  - e is an isolated vertex of G_4
- If both endpoints of an edge are the same vertex, then we say the edge is a ***loop***.
  - bb is a loop in G_4
- If there is more than one edge with the same endpoints, we call these ***multi-edges***.
  - cd is a multi-edge of G_4
- If a graph has no multi-edges or loops, we call it ***simple***.
- The ***degree*** of a vertex v, denoted $\deg(v)$, is the number of edges incident to v, with a loop adding two to the degree. If the degree is even, the vertex is called ***even***; if the degree is odd, then the vertex is ***odd***.
  - $\deg(a) = 2$, $\deg(b) = 4$, $\deg(c) = 3$, $\deg(d) = 3$, $\deg(e) = 0$
- If all vertices in a graph G have the same degree k, then G is called a **k-regular** graph. When $k = 3$, we call the graph **cubic**.

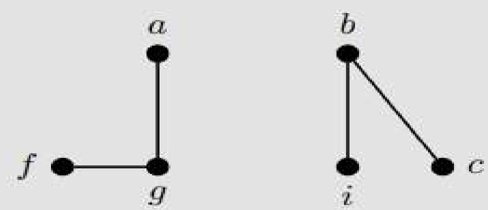
Example 1.3 Consider the graph G below. Find two subgraphs of G, both of which have vertex set $V' = \{a, b, c, f, g, i\}$.



Solution: Two possible solutions are shown below. Note that the graph H_1 on the left contains every edge from G amongst the vertices in V', whereas the graph H_2 on the right does not since some of the available edges are missing (namely, ab, af, ci, and gi).



H_1



H_2

The graph shown on the left above is a special type, called an induced subgraph, since all the edges are present between the chosen vertices. Another special type of subgraph, called a spanning subgraph, includes all the vertices of the original graph.

Definition 1.5 Given a graph $G = (V, E)$, an ***induced subgraph*** is a subgraph $G[V']$ where $V' \subseteq V$ and every available edge from G between the vertices in V' is included.

We say H is a ***spanning subgraph*** if it contains all the vertices but not necessarily all the edges of G; that is, $V(H) = V(G)$ and $E(H) \subseteq E(G)$.

Definition 1.8 A ***weighted graph*** $G = (V, E, w)$ is a graph where each of the edges has a real number associated with it. This number is referred to as the ***weight*** and denoted $w(xy)$ for the edge xy.

See Example # 1.8

Definition 1.9 A simple graph G is ***complete*** if every pair of distinct vertices is adjacent. The complete graph on n vertices is denoted K_n.

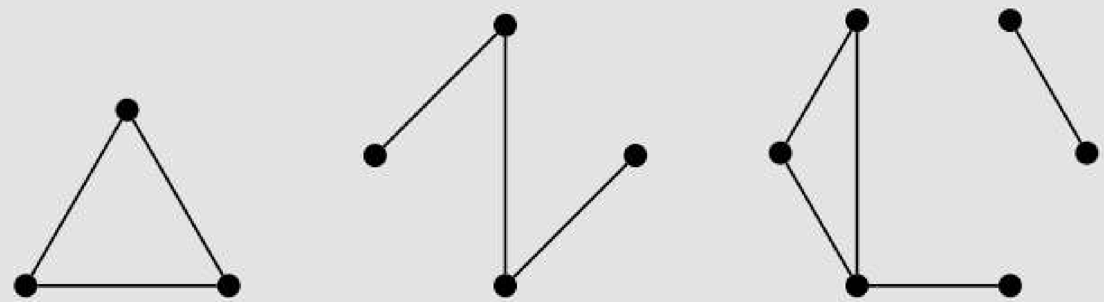
Properties of K_n

(1) Each vertex in K_n has degree $n - 1$.

(2) K_n has $\dfrac{n(n-1)}{2}$ edges.

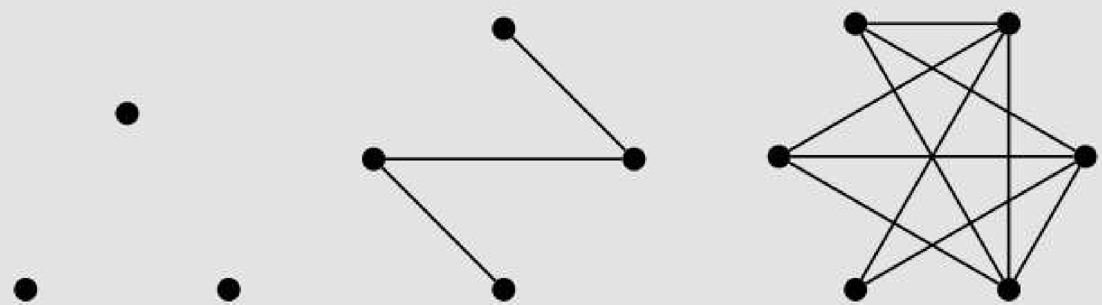
(3) K_n contains the most edges out of all simple graphs on n vertices.

Example 1.10 Find the complements of each graph shown below.



G_1 $\quad$ G_2 $\quad$ G_3

Solution: For each graph we simply add an edge where there wasn't one before and remove the current edges.



$\overline{G_1}$ $\quad$ $\overline{G_2}$ $\quad$ $\overline{G_3}$

Definition 1.12 A graph G is ***bipartite*** if the vertices can be partitioned into two sets X and Y so that every edge has one endpoint in X and the other in Y.

Definition 1.13 $K_{m,n}$ is the ***complete bipartite graph*** where $|X| = m$ and $|Y| = n$ and every vertex in X is adjacent to every vertex in Y.

NOTE:

A **directed graph** or **digraph** consists of a finite set V of vertices and *a set A* of *ordered pairs* of distinct vertices called **arcs.** If the ordered pair $\{u, v\}$ is an arc a, we say that the arc a is **directed from** u to v. In this context, arc a is **adjacent from** vertex u and is **adjacent to** vertex v. In a **mixed graph,** there will be at least one edge and at least one arc. If each arc of a digraph is replaced by an edge, the resulting structure is a graph known as the **underlying graph** of the digraph. On the other hand, if each edge of a simple graph is replaced by an arc, the resulting structure is a digraph known as an **orientation** of the simple graph. Any orientation of a complete graph is known as a **tournament.**

SUMMARY OF WEEK # 01

A **graph** G consists of a set V of **vertices** and a collection E (not necessarily a set) of unordered pairs of vertices called **edges.** A graph is symbolically represented as $G = (V, E)$. In this book, unless otherwise specified, both V and E are finite. The **order** of a graph is the number of its vertices, and its **size** is the number of its edges. If u and v are two vertices of a graph and if the *unordered pair* $\{u, v\}$ is an edge denoted by e, we say that e **joins** u and v or that it is an edge between u and v. In this case, the vertices u and v are said to be **incident on** e and e is **incident to** both u and v. Two or more edges that join the same pair of distinct vertices are called **parallel** edges. An edge represented by an unordered pair in which the two elements are not distinct is known as a **loop.** A graph with no loops is a **multigraph.** A graph with at least one loop is a **pseudograph.** A **simple graph** is a graph with no parallel edges and loops. The term *graph* is used in lieu of *simple graph* in many places in this book. The **complete graph** K_n is a graph with n vertices in which there is exactly one edge joining every pair of vertices. The graph K_1 with one vertex and no edge is known as the **trivial graph.** A **bipartite graph** is a simple graph in which the set of vertices can be partitioned into two sets X and Y such that every edge is between a vertex in X and a vertex in Y; it is represented as $G = (X, Y, E)$. The **complete bipartite graph** $K_{m,n}$ is the graph (X, Y, E) with m vertices in X and n vertices in Y in which there is an edge between every vertex in X and every vertex in Y. The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph $G = G_1 \cup G_2 = (V, E)$, where V is the union of V_1 and V_2 and E is the union of E_1 and E_2.

EX #: 1.8

Problems:

1.1-1.7, 1.12,1.14, 1.15, 1.16, 1.17, 1.20, 1.22