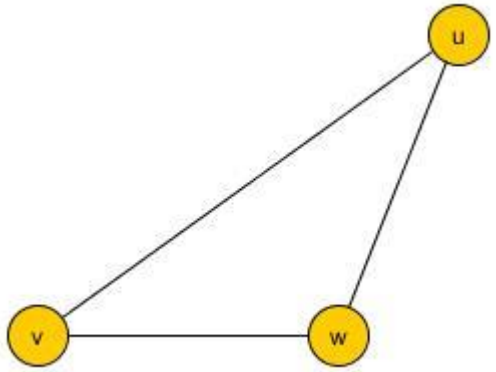
Graph Terminology and Special Types of Graphs

DEFINITION 1 Two vertices *u* and *v* in an undirected graph *G* are called adjacent (смежные) (or neighbors) in *G* if *u* and *v* are endpoints of an edge *e* of *G*.

Such an edge *e* is called incident with (инцидентно) the vertices *u* and *v*

and *e* is said to connect (соединяет) *u* and *v*.

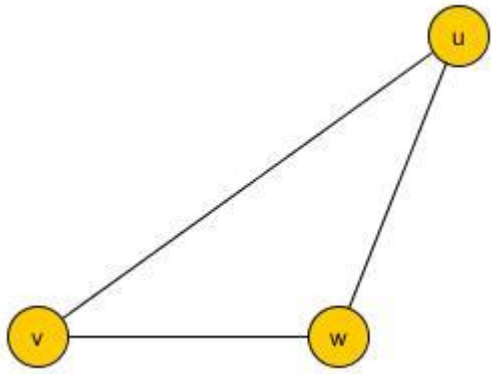


DEFINITION 2 The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the neighborhood (окружение) of v.

If A is a subset of V , we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A.

So,

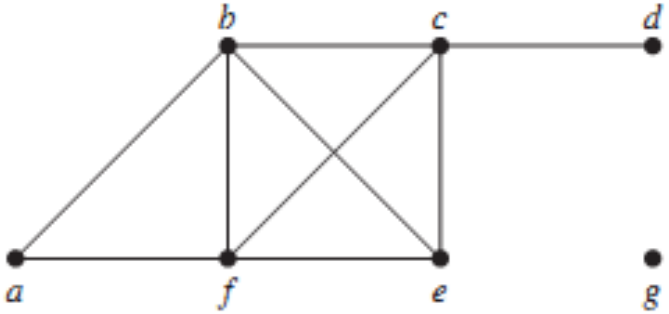
$$N(A) = \bigcup_{v \in A} N(v)$$



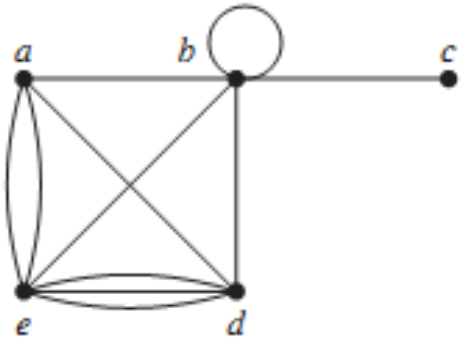
DEFINITION 3 The degree (степень) of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.

The degree of the vertex v is denoted by $\deg(v)$.

EXAMPLE What are the degrees and what are the neighborhoods of the vertices in the graphs G and H?



G



H

Solution: In G, $\deg(a) = 2$, $\deg(b) = \deg(c) = \deg(f) = 4$, $\deg(d) = 1$, $\deg(e) = 3$, and $\deg(g) = 0$.
The neighborhoods of these vertices are $N(a) = \{b, f\}$, $N(b) = \{a, c, e, f\}$, $N(c) = \{b, d, e, f\}$, $N(d) = \{c\}$, $N(e) = \{b, c, f\}$, $N(f) = \{a, b, c, e\}$, and $N(g) = \emptyset$.

In H, $\deg(a) = 4$, $\deg(b) = \deg(e) = 6$, $\deg(c) = 1$, and $\deg(d) = 5$.
The neighborhoods of these vertices are $N(a) = \{b, d, e\}$, $N(b) = \{a, b, c, d, e\}$, $N(c) = \{b\}$, $N(d) = \{a, b, e\}$, and $N(e) = \{a, b, d\}$.

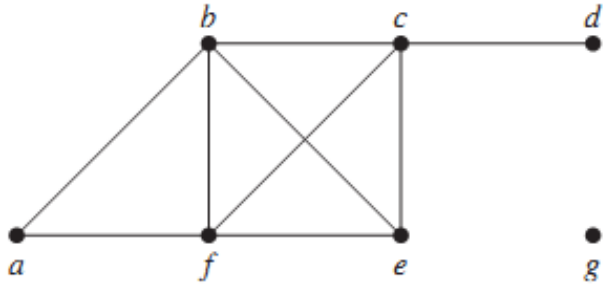
A vertex of degree zero is called isolated (изолированная).

It follows that an isolated vertex is not adjacent to any vertex.

Vertex *g* in graph *G* is isolated.

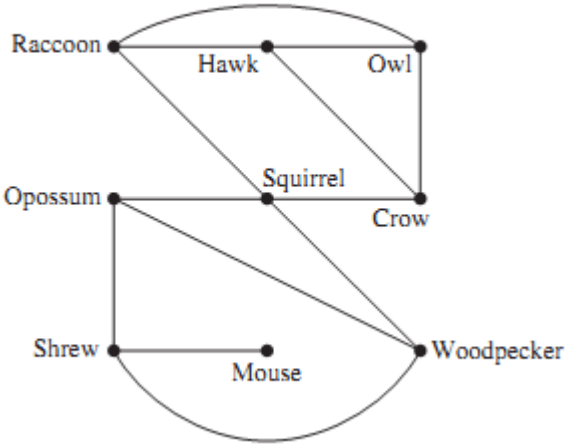
A vertex is pendant (висячая) ⇔ it has degree 1.
Consequently, a pendant vertex is adjacent to exactly 1 other vertex.

Vertex *d* in graph *G* bellow is pendant.



G

EXAMPLE 2



What does the degree of a vertex in a niche overlap graph represent?
Which vertices in this graph are pendant and which are isolated?
Use the niche overlap graph to interpret your answers.
Solution: There is an edge between two vertices in a niche overlap graph ⇔the two species represented by these vertices compete.
Hence, the degree of a vertex in a niche overlap graph is the number of species in the ecosystem that compete with the species represented by this vertex.
A vertex is pendant if the species competes with exactly one other species in the ecosystem.
Finally, the vertex representing a species is isolated if this species does not compete with any other species in the ecosystem

What do we get when we add the degrees of all the vertices of a graph $G = (V, E)$?

Each edge contributes 2 to the sum of the degrees of the vertices because an edge is incident with exactly 2 (possibly equal) vertices.

This means that the sum of the degrees of the vertices is twice the number of edges.

The handshaking theorem

Let $G = (V, E)$ be an undirected graph with m edges.

Then

$$2m = \sum_{v \in V} \deg(v).$$

(Note that this applies even if multiple edges and loops are present.)

EXAMPLE

How many edges are there in a graph with 10 vertices each of degree 6?

Solution: Because the sum of the degrees of the vertices is $6 \cdot 10 = 60$, it follows that $2m = 60$, where m is the number of edges.

Therefore, $m = 30$.

Theorem 1 shows that the sum of the degrees of the vertices of an undirected graph is even.

This simple fact has many consequences, one of which is given as Theorem 2.

THEOREM 2 An undirected graph has an even number of vertices of odd degree.

Proof: Let V_1 be the set of vertices of even degree and V_2 the set of vertices of odd degree in an undirected graph G = (V,E) with m edges.

Then

$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v).$$

Because $deg(v)$ is even for $v \in V_1$, the first term in the right-hand side of the last equality is even.

Furthermore, the sum of the two terms on the right-hand side of the last equality is even, because this sum is $2m$.

Hence, the second term in the sum is also even.

Because all the terms in this sum are odd, there must be an even number of such terms.

Thus, there are an even number of vertices of odd degree.

Terminology for graphs with directed edges

DEFINITION 4 When (u, v) is an edge of the graph G with directed edges, u is said to be adjacent to (смежная) v and v is said to be adjacent from (смежная) u.

The vertex u is called the initial (начальная) vertex of (u, v), and v is called the terminal or end (конечная) vertex of (u, v).

The initial vertex and terminal vertex of a loop are the same.

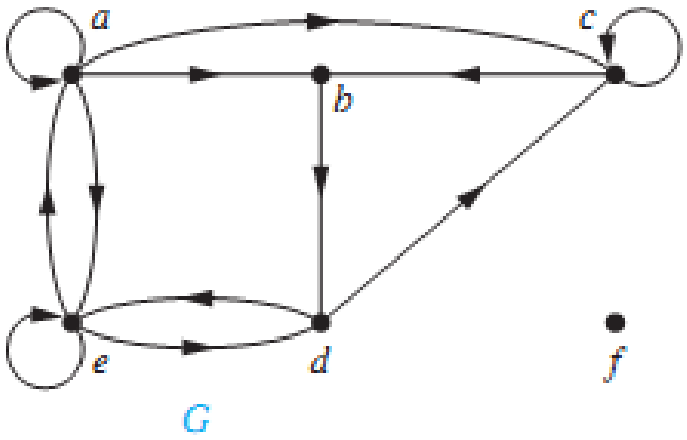
Because the edges in graphs with directed edges are ordered pairs, the definition of the degree of a vertex can be refined to reflect the number of edges with this vertex as the initial vertex and as the terminal vertex.

DEFINITION 5 In a graph with directed edges the in-degree of a vertex (полустепень захода) v, denoted by $\deg^-(v)$, is the number of edges with v as their terminal vertex.

The out-degree of (полустепень исхода) v, denoted by $\deg^+(v)$, is the number of edges with v as their initial vertex.

(Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

EXAMPLE 4 Find the in-degree and out-degree of each vertex in the graph G with directed edges



Solution: The in-degrees in G are $\deg^-(a) = 2$, $\deg^-(b) = 2$, $\deg^-(c) = 3$, $\deg^-(d) = 2$, $\deg^-(e) = 3$, and $\deg^-(f) = 0$.

The out-degrees are $\deg^+(a) = 4$, $\deg^+(b) = 1$, $\deg^+(c) = 2$, $\deg^+(d) = 2$, $\deg^+(e) = 3$, and $\deg^+(f) = 0$.

THEOREM 3 Let $G = (V, E)$ be a graph with directed edges. Then

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|.$$

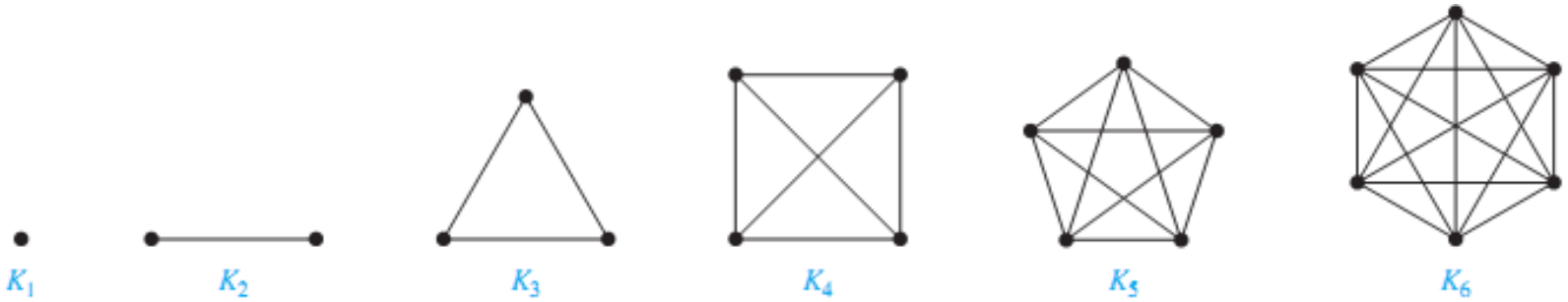
There are many properties of a graph with directed edges that do not depend on the direction of its edges.
Consequently, it is often useful to ignore these directions.
The undirected graph that results from ignoring directions of edges is called the underlying undirected graph (основание).
A graph with directed edges and its underlying undirected graph have the same number of edges.

Some Special Simple Graphs

We will now introduce several classes of simple graphs.

These graphs are often used as examples and arise in many applications.

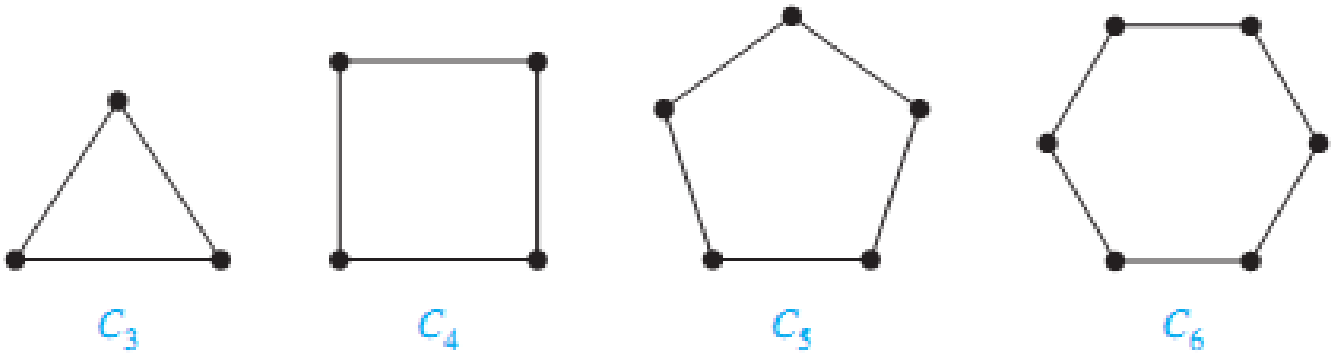
Complete Graphs (Полные графы)



A complete graph on *n* vertices, denoted by K_n, is a simple graph that contains exactly 1 edge between each pair of distinct vertices. The graphs K_n, for $n = 1, 2, 3, 4, 5, 6$, are displayed .

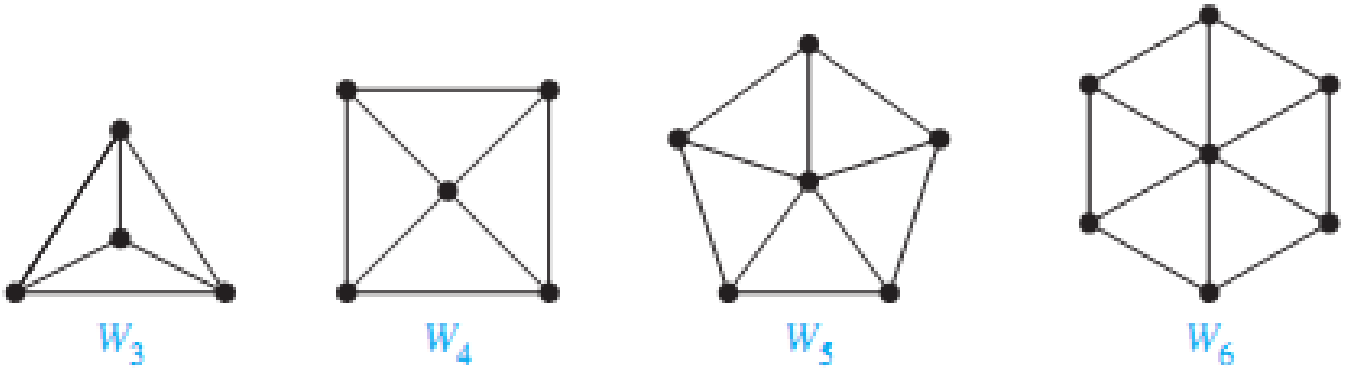
A simple graph for which there is at least 1 pair of distinct vertex not connected by an edge is called noncomplete.

Cycles (Простые циклы)



A **cycle C_n**, $n \geq 3$, consists of n vertices v_1, v_2,..., v_n and edges $\{v_1, v_2\}$, $\{v_2, v_3\}$,..., $\{v_{n-1}, v_n\}$, and $\{v_n, v_1\}$.
The cycles C_3, C_4, C_5, and C_6 are displayed.

Wheels (Колеса)



We obtain a wheel W_n when we add an additional vertex to a cycle C_n, for $n \geq 3$,
and connect this new vertex to each of the n vertices in C_n, by new edges.

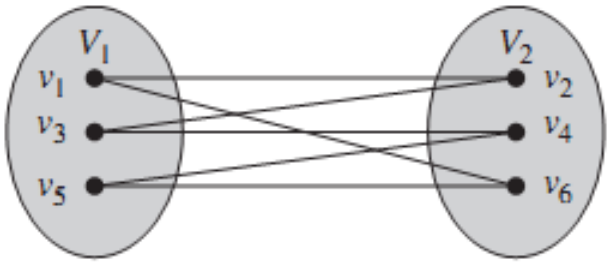
The wheels W_3, W_4, W_5, and W_6 are displayed.

Bipartite graphs (двудольные графы)

A simple graph G is called bipartite if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2

(so that no edge in G connects either 2 vertices in V_1 or 2 vertices in V_2).

When this condition holds, we call the pair (V_1, V_2) a bipartition of the vertex set V of G.



$V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4, v_6\}$, and every edge of C_6 connects a vertex in V_1 and a vertex in V_2.

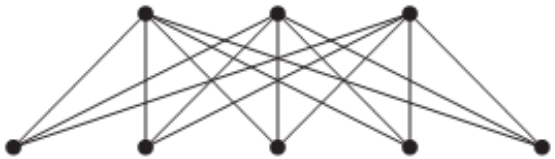
Complete Bipartite Graphs

A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into 2 subsets of *m* and *n* vertices, respectively with an edge between 2 vertices ⇔ one vertex is in the first subset and the other vertex is in the second subset.

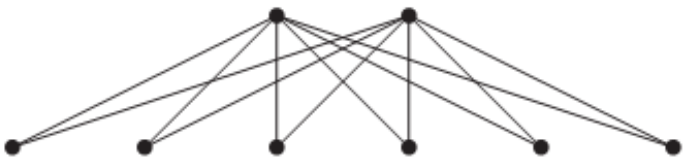
The complete bipartite graphs $K_{2,3}$, $K_{3,3}$, $K_{3,5}$, and $K_{2,6}$ are displayed bellow.

$K_{2,3}$

$K_{3,3}$

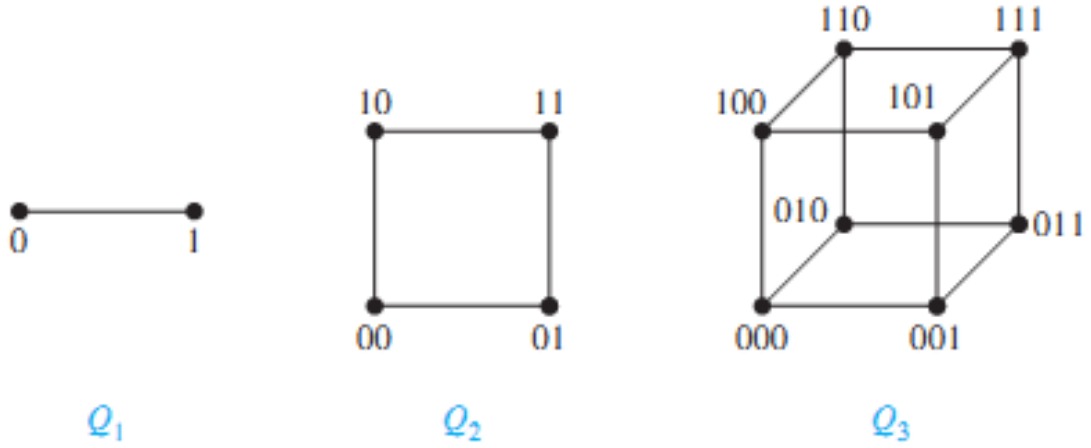


$K_{3,5}$



$K_{2,6}$

n-Cubes (*n*- мерные кубы)



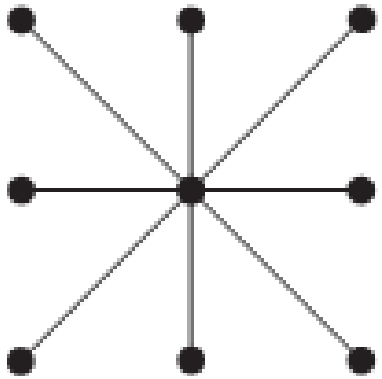
An n-dimensional hypercube, or n-cube, denoted by Q_n, is a graph that has vertices representing the 2^n bit strings of length *n*.

2 vertices are adjacent ⇔ the bit strings that they represent differ in exactly 1 bit position.

We display Q_1, Q_2, and Q_3.

Note that you can construct the (n + 1)-cube Q_{n+1} from the n-cube Q_n by making 2 copies of Q_n, prefacing the labels on the vertices with a 0 in one copy of Q_n and with a 1 in the other copy of Q_n, and adding edges connecting two vertices that have labels differing only in the first bit.

Local Area Networks (LAN)



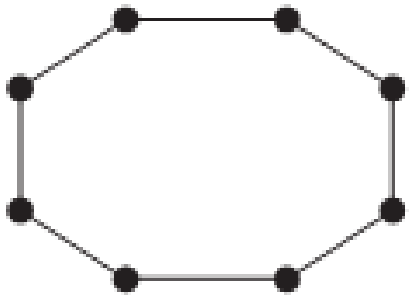
The various computers in a building, such as minicomputers and personal computers, as well as peripheral devices such as printers and plotters, can be connected using a local area network.

Some of these networks are based on a star topology, where all devices are connected to a central control device.

A local area network can be represented using a complete bipartite graph $K_{1,n}$, .

Messages are sent from device to device through the central control device.

Ring topology

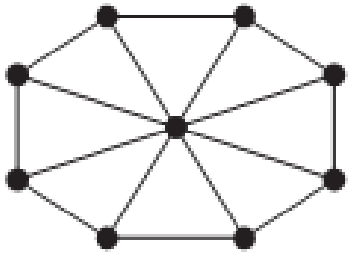


Other local area networks are based on a ring topology, where each device is connected to exactly 2 others.

Local area networks with a ring topology are modeled using *n*-cycles, C_n.

Messages are sent from device to device around the cycle until the intended recipient of a message is reached.

Wheel topology



Some local area networks use a hybrid of these two topologies.

Messages may be sent around the ring, or through a central device.

This redundancy makes the network more reliable.

Local area networks with this redundancy can be modeled using wheels W_n.

Interconnection Networks for Parallel Computation

For many years, computers executed programs 1 operation at a time.

Consequently, the algorithms written to solve problems were designed to perform 1 step at a time; such algorithms are called serial (последовательные).

However, many computationally intense problems, such as weather simulations, medical imaging, and cryptanalysis, cannot be solved in a reasonable amount of time using serial operations, even on a supercomputer.

Furthermore, there is a physical limit to how fast a computer can carry out basic operations, so there will always be problems that cannot be solved in a reasonable length of time using serial operations.

Parallel processing, which uses computers made up of many separate processors, each with its own memory, helps overcome the limitations of computers with a single processor.

Parallel algorithms, which break a problem into a number of subproblems that can be solved concurrently, can then be devised to rapidly solve problems using a computer with multiple processors.

In a parallel algorithm, a single instruction stream controls the execution of the algorithm, sending subproblems to different processors, and directs the input and output of these subproblems to the appropriate processors.

When parallel processing is used, one processor may need output generated by another processor.

Consequently, these processors need to be interconnected.

We can use the appropriate type of graph to represent the interconnection network of the processors in a computer with multiple processors.

In the following discussion, we will describe the most commonly used types of interconnection networks for parallel processors.

The type of interconnection network used to implement a particular parallel algorithm depends on the requirements for exchange of data between processors, the desired speed, and, of course, the available hardware.

Complete graph on *n* vertices

The simplest, but most expensive, network-interconnecting processors include a 2-way link between each pair of processors.

This network can be represented by K_n, the complete graph on *n* vertices, when there are *n* processors.

However, there are serious problems with this type of interconnection network because the required number of connections is so large.

In reality, the number of direct connections to a processor is limited, so when there are a large number of processors, a processor cannot be linked directly to all others.

For example, when there are 64 processors, C(64, 2) = 2016 connections would be required, and each processor would have to be directly connected to 63 others.

A Linear Array of processors.

On the other hand, perhaps the simplest way to interconnect n processors is to use an arrangement known as a linear array.

Each processor P_i , other than P_1 and P_n, is connected to its neighbors P_{i-1} and P_{i+1} via a 2-way link.

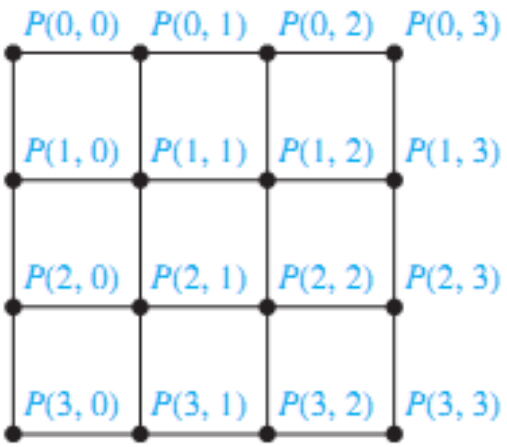
P_1 is connected only to P_2, and P_n is connected only to P_{n-1}.

The linear array for 6 processors is shown above.

The advantage of a linear array is that each processor has at most 2 direct connections to other processors.

The disadvantage is that it is sometimes necessary to use a large number of intermediate links, called hops, for processors to share information.

Mesh network



A Mesh Network for
16 Processors.

The mesh network (or two-dimensional array) is a commonly used interconnection network.

In such a network, the number of processors is a perfect square, say $n = m^2$.

The *n* processors are labeled *P*(i,j), $0 \le i \le m - 1$, $0 \le j \le m - 1$.

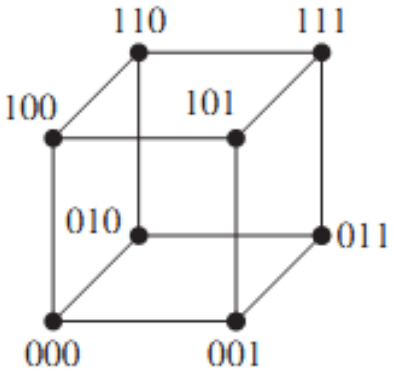
2-way links connect processor *P*(i,j) with its four neighbors, processors $P(i \pm 1,j)$ and $P(i,j \pm 1)$, as long as these are processors in the mesh.

(Note that four processors, on the corners of the mesh, have only 2 adjacent processors, and other processors on the boundaries have only 3 neighbors.

Sometimes a variant of a mesh network in which every processor has exactly *4* connections is used.

The mesh network limits the number of links for each processor. Communication between some pairs of processors requires $O(\sqrt{n}) = O(m)$ intermediate links

Hypercube network



One important type of interconnection network is the hypercube.

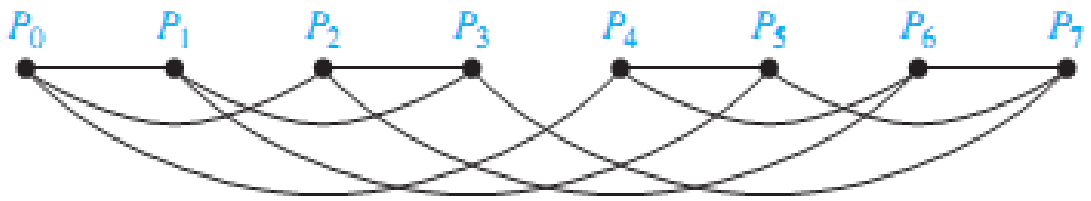
For such a network, the number of processors is a power of 2, $n = 2^m$.

The n processors are labeled $P_0, P_1, \ldots, P_{n-1}$.

Each processor has 2-way connections to m other processors.

Processor P_i is linked to the processors with indices whose binary representations differ from the binary representation of i in exactly 1 bit.

Hypercube network

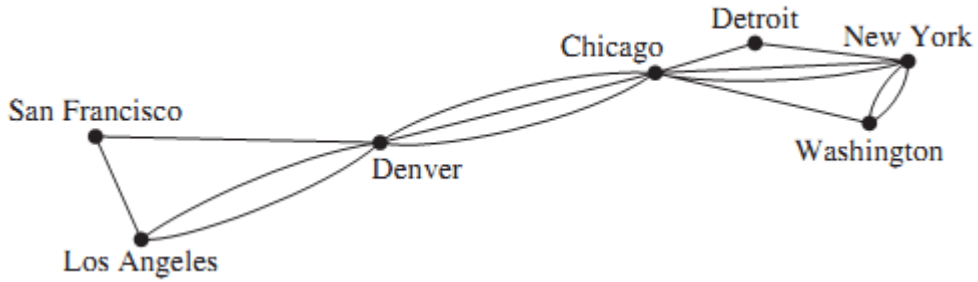


The hypercube network balances the number of direct connections for each processor and the number of intermediate connections required so that processors can communicate.

Many computers have been built using a hypercube network, and many parallel algorithms have been devised that use a hypercube network.

The graph Q_m, the m-cube, represents the hypercube network with $n = 2^m$ processors.

New Graphs from Old



Sometimes we need only part of a graph to solve a problem.

For instance, we may care only about the part of a large computer network that involves the computer centers in New York, Denver and Detroit .

Then we can ignore the other computer centers and all telephone lines not linking 2 of these specific 3 computer centers.

In the graph model for the large network, we can remove the vertices corresponding to the computer centers other than the 3 of interest, and we can remove all edges incident with a vertex that was removed.

When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained.

Such a graph is called a subgraph of the original graph.

DEFINITION 7

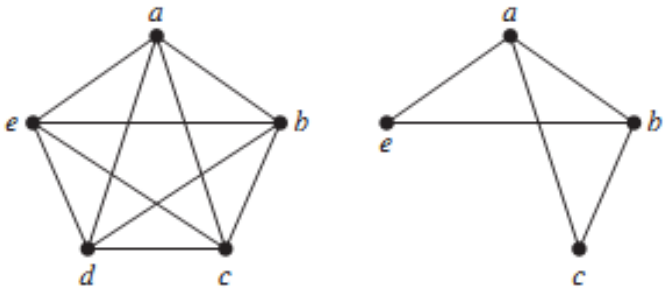
A *subgraph of a graph (подграф графа)* $G = (V, E)$ is a graph $H = (W, F)$, where $W \subseteq V$ and $F \subseteq E$.

A subgraph H of G is a proper subgraph (собственный подграф) of G if $H \neq G$.

Given a set of vertices of a graph, we can form a subgraph of this graph with these vertices

and the edges of the graph that connect them.

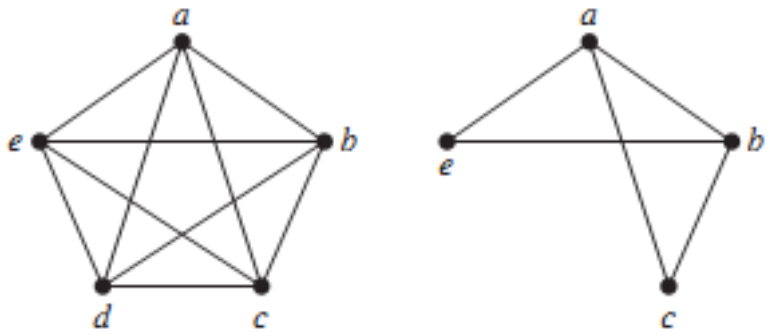
The graph G is a subgraph of K_5.



DEFINITION 8

Let $G = (V, E)$ be a simple graph.

The subgraph induced (индуцированный подграф) by a subset W of the vertex set V is the graph (W, F), where the edge set F contains an edge in $E \Leftrightarrow$ both endpoints of this edge are in W.



If we add the edge $\{c, e\}$ to G, we obtain the subgraph induced by $W = \{a, b, c, e\}$.

REMOVING EDGES OF A GRAPH

Given a graph $G = (V, E)$ and an edge $e \in E$, we can produce a subgraph of G by removing the edge e.

The resulting subgraph, denoted by $G - e$, has the same vertex set V as G.

Its edge set is $E - e$.

Hence, $G - e = (V, E - \{e\})$.

Similarly, if E' is a subset of E, we can produce a subgraph of G by removing the edges in E' from the graph.

The resulting subgraph has the same vertex set V as G.

Its edge set is $E - E'$.

ADDING EDGES TO A GRAPH

We can also add an edge e to a graph to produce a new larger graph when this edge connects two vertices already in G.

We denote by $G + e$ the new graph produced by adding a new edge e, connecting two previously nonadjacent vertices, to the graph G

Hence, $G + e = (V, E \cup \{e\})$.

The vertex set of $G + e$ is the same as the vertex set of G and the edge set is the union of the edge set of G and the set $\{e\}$.

REMOVING VERTICES FROM A GRAPH

When we *remove a vertex v* and all edges incident to it from $G = (V,E)$, we produce a subgraph, denoted by $G - v$.

Observe that $G - v = (V - v, E')$, where E' is the set of edges of G not incident to v.

Similarly, if V' is a subset of V , then the graph $G - V'$ is the subgraph $(V - V', E')$, where E' is the set of edges of G not incident to a vertex in V'.

GRAPH UNIONS ($G_1 \cup G_2$)

2 or more graphs can be combined in various ways.

The new graph that contains all the vertices and edges of these graphs is called the union of the graphs.

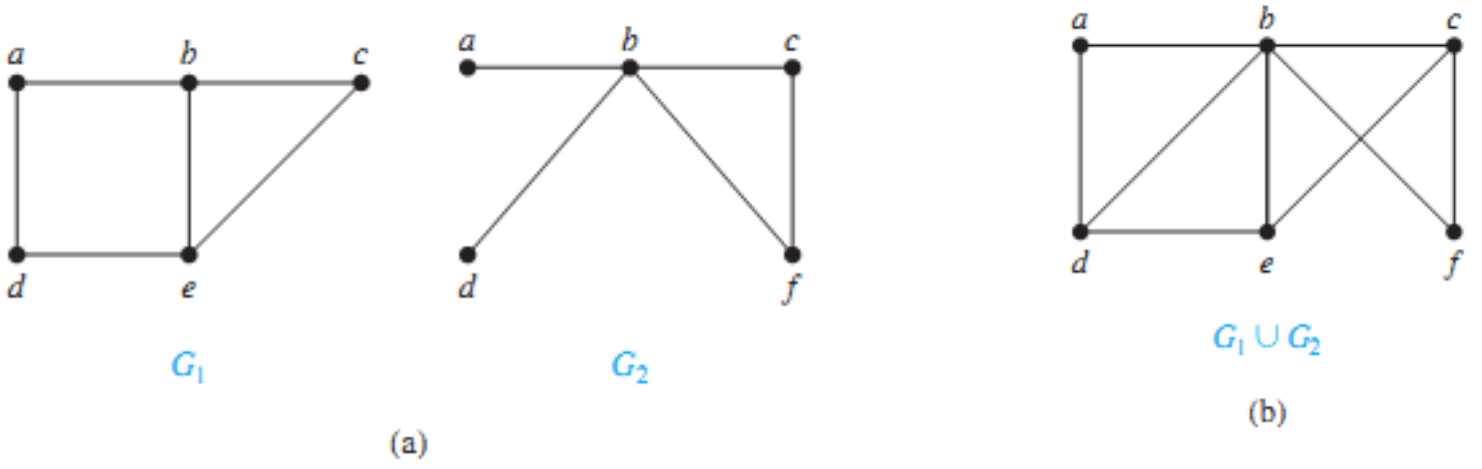
The union of 2 simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.

The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.

EXAMPLE Find the union of the graphs G_1 and G_2

Solution: The vertex set of the union $G_1 \cup G_2$ is the union of the two vertex sets, namely, {*a, b, c, d, e, f* }.

The edge set of the union is the union of the 2 edge sets.



(a) The Simple Graphs G_1 and G_2;

(b) Their Union $G_1 \cup G_2$.

Intersection of graphs ($G_1 \cap G_2$)

The intersection of 2 simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cap V_2$ and edge set $E_1 \cap E_2$.

The intersection of G_1 and G_2 is denoted by $G_1 \cap G_2$.

Complementary graph

The complementary graph *G'* of a simple graph *G* has the same vertices as *G*.

2 vertices are adjacent in *G'* ⇔ they are not adjacent in *G*.