C:\Documents and Settings\admin\Desktop\defEx_files\redline.gif

**Definitions and Examples**

C:\Documents and Settings\admin\Desktop\defEx_files\redline.gif

Informally, a graph is a diagram consisting of points, called vertices, joined together by lines, called edges; each edge joins exactly two vertices. A graph G is a triple consisting of a vertex set of V(*G*), an edge set E(G), and a relation that associates with each edge two vertices (not necessarily distinct) called its endpoints.

**Definition of Graph**

 A graph G = (V, E) consists of a (finite) set denoted by V, or by V(G) if one wishes to make clear which graph is under consideration, and a collection E, or E(G), of unordered pairs {u, v} of distinct elements from V. Each element of V is called a vertex or a point or a node, and each element of E is called an edge or a line or a link.

Formally, a graph G is an ordered pair of dsjoint sets (V, E), where E V × V. Set V is called the vertex or node set, while set E is the edge set of graph G. Typically, it is assumed that self-loops (i.e. edges of the form (u, u), for some u  V) are not contained in a graph.

**Directed and Undirected Graph**

A graph G = (V, E) is directed if the edge set is composed of ordered vertex (node) pairs. A graph is undirected if the edge set is composed of unordered vertex pair.

**Vertex Cardinality**

The number of vertices, the cardinality of V, is called the order of graph and devoted by |V|. We usually use *n* to denote the order of G. The number of edges, the cardinality of E, is called the size of graph and denoted by |E|. We usually use *m* to denote the size of G.

**Neighbor Vertex and Neighborhood**

We write vivj E(G) to mean {vi, vj} E(G), and if e = vi vj E(G), we say vi and vj are adjacent.

Formally, given a graph G = (V, E), two vertices  vi , vjV are said to be neighbors, or adjacent nodes, if ( vi , vj) E. If G is directed, we distinguish between incoming neighbors of vi (those vertices vj V such that (vj, vi) E) and outgoing neighbors of vi (those vertices vjV such that (vi, vj) E).

The open neighborhood N(v) of the vertex v consists of the set vertices adjacent to v, that is, N(v) = {w v : vw E}. The closed neighborhood of v is N[v] = N(v){v}. For a set S V, the open neighborhood N(S) is defined to be UvSN(v), and the closed neighborhood of S is N[S] = N(S)S.

**Vertex Degree**

The degree deg(v) of vertex v is the number of edges incident on v or equivalently, deg(v) = |N(v)|. The degree sequence of graph is (deg(v1), deg(v2), ..., deg(vn)), typically written in nondecreasing or nonincreasing order. The minimum and maximum degree of vertices in V(G) are denoted by (G) and ∆(G), respectively. If (G) = ∆(G) = r, then graph G is said to be regular of degree r, or simply r-regular.

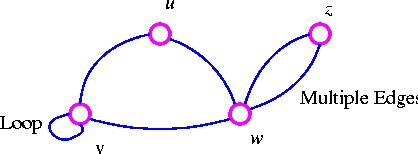
Formally, given a graph G = (V, E), the degree of a vertex v V is the number of its neighbors in the graph. That is,

                                            deg(v) = | {u V : (v, w) E}|.

If G is directed, we distinguish between in-degree (nimber of incoming neighbors) and out-degree (number of outgoing neighbors) of a vertex.

**Loop and Multiple Edges**

A loop is an edge whose endpoints are equal i.e., an edge joining a vertex to it self is called a loop. We say that the graph has multiple edges if in the graph two or more edges joining the same pair of vertices.



**Simple Graph**

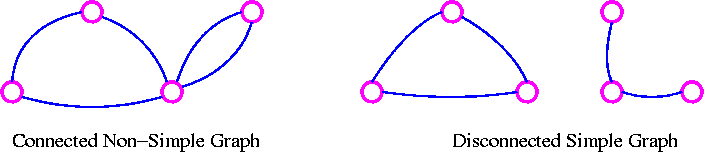
A graph with no loops or multiple edges is called a simple graph. We specify a simple graph by its set of vertices and set of edges, treating the edge set as a set of unordered pairs of vertices and write *e* = *uv* (or *e* = *vu*) for an edge *e* with endpoints *u* and *v*.

C:\Documents and Settings\admin\Desktop\defEx_files\g2.gif

When *u* and *v* are endpoints of an edge, they are adjacent and are neighbors.

**Connected Graph**

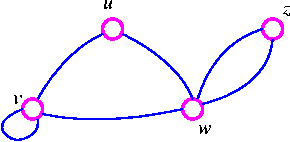
A graph that is in one piece is said to be connected, whereas one which splits into several pieces is disconnected.



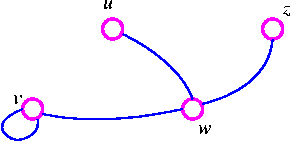
A graph G is connected if there is a path in G between any given pair of vertices, otherwise it is disconnected. Every disconnected graph can be split up into a number of connected subgraphs, called components.

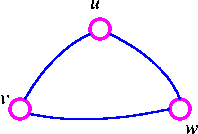
**Subgraph**

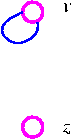
Let *G* be a graph with vertex set *V*(*G*) and edge-list E(G). A subgraph of *G* is a graph all of whose vertices belong to *V*(*G*) and all of whose edges belong to *E*(*G*). For example, if *G*is the connected graph below:



where *V*(*G*) = {*u*,*v*, *w*, *z*} and*E*(*G*) = (*uv*, *uw*, *vv*, *vw*, *wz*, *wz*} then the following four graphs are subgraphs of G.



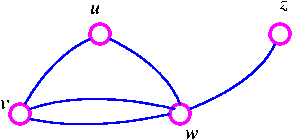




**Degree (or Valency)**

Let *G* be a graph with loops, and let *v* be a vertex of *G*. The degree of *v* is the number of edges meeting at *v*, and is denoted by *deg*(*v*).

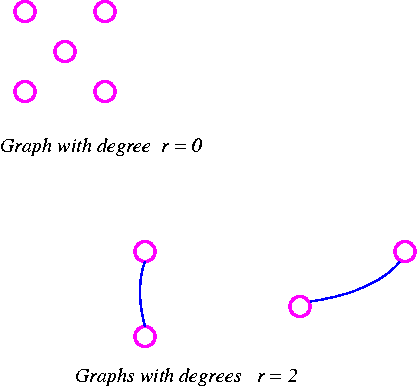
For example, consider, the following graph G

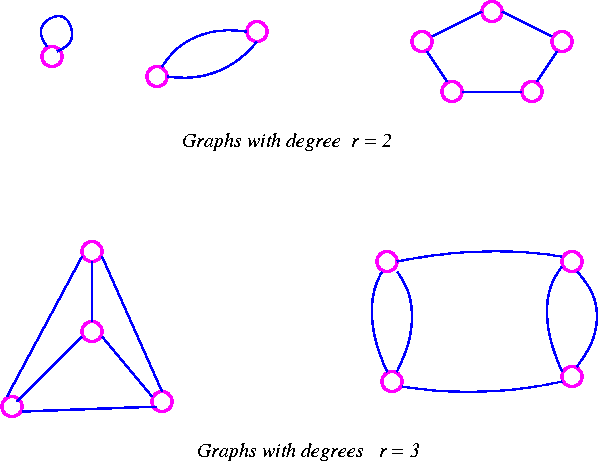


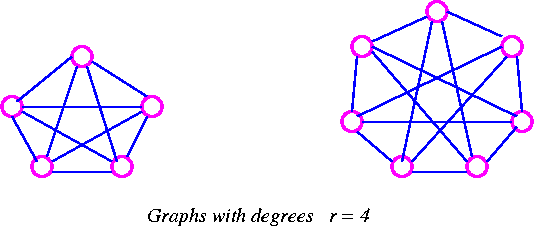
The graph *G* has *deg*(*u*) = 2, *deg*(*v*) = 3, *deg*(w) = 4 and *deg*(*z*) = 1.

**Regular Graph**

A graph is regular if all the vertices of *G* have the same degree. In particular, if the degree of each vertex is *r*, the *G* is regular of degree *r*.







**The Handshaking Lemma***In any graph, the sum of all the vertex-degree is equal to twice the number of edges*.

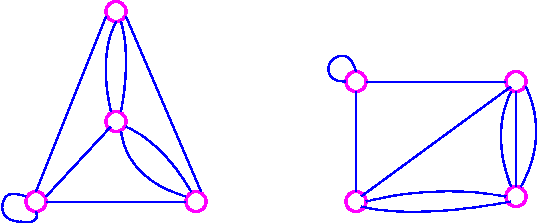
***Proof***Since each edge has two ends, it must contribute exactly 2 to the sum of the degrees. The result follows immediately.

The Following are the consequences of the Handshaking lemma.

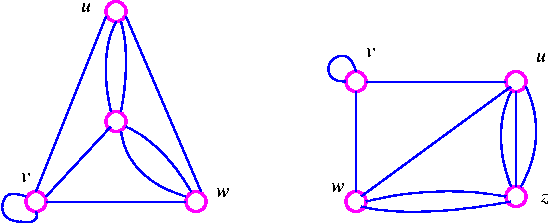
1. In any graph, the sum of all the vertex-degree is an even number.
2. In any graph, the number of vertices of odd degree is even.
3. If G is a graph which has n vertices and is regular of degree r, then G has exactly 1/2 nr edges.

**Isomorphic Graphs**

Two graph G and H are isomorphic if H can be obtained from G by relabeling the vertices - that is, if there is a one-to-one correspondence between the vertices of G and those of H, such that the number of edges joining any pair of vertices in G is equal to the number of edges joining the corresponding pair of vertices in H. For example, two unlabeled graphs, such as



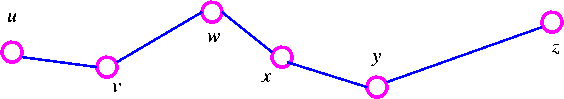
are isomorphic if labels can be attached to their vertices so that they become the same graph.



The word isomorphic derives from the Greek for same and form.

**Walk**

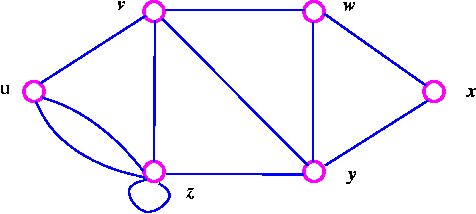
A walk of length k in a graph *G* is a succession of *k* edges of G of the form *uv*, *vw*, *wx*, . . . , *yz*.



We denote this walk by *uvwx* . . *yz* and refer to it as a walk between *u* and *z*.

**Trail and Path**

If all the edges (but no necessarily all the vertices) of a walk are different, then the walk is called a trail. If, in addition, all the vertices are difficult, then the trail is called path.

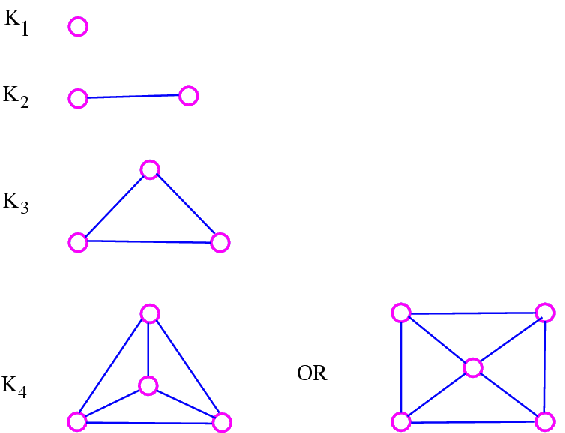


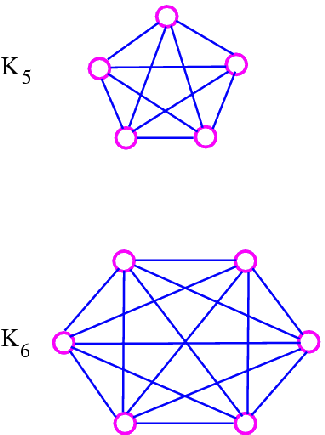
* The walk *vzzywxy*is a trail since the vertices y and z both occur twice.
* The walk *vwxyz*is a path since the walk has no repeated vertices.

**Complete Graphs**

A computer graph is a graph in which every two distinct vertices are joined by exactly one edge. The complete graph with *n* vertices is denoted by  K*n*.

The following are the examples of complete graphs.



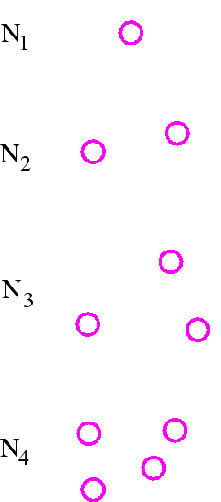


The graph K*n*is regular of degree *n*-1, and therefore has 1/2*n*(*n*-1) edges, by consequence 3 of the handshaking lemma.

**Null Graphs**

A null graphs is a graph containing no edges. The null graph with n vertices is denoted by N*n*.

The following are the examples of null graphs.

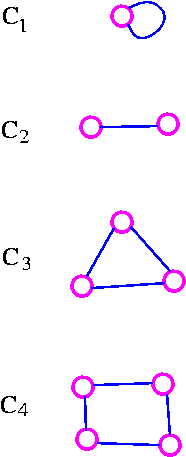


Note that N*n*is regular of degree 0.

**Cycle Graphs**

A cycle graph is a graph consisting of a single cycle. The cycle graph with n vertices is denoted by C*n*.

The following are the examples of cyclic graphs.

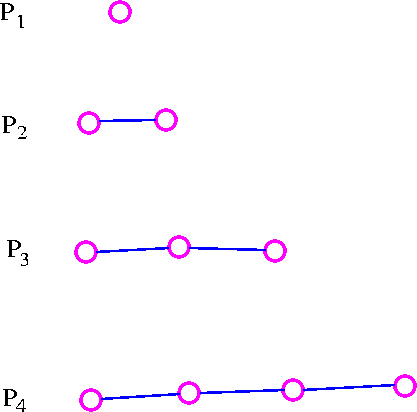


Note that  C*n*is regular of degree 2, and has *n* edges.

**Path Graphs**

A path graph is a graph consisting of a single path. The path graph with n vertices is denoted by P*n*.

The following are the examples of path graphs.

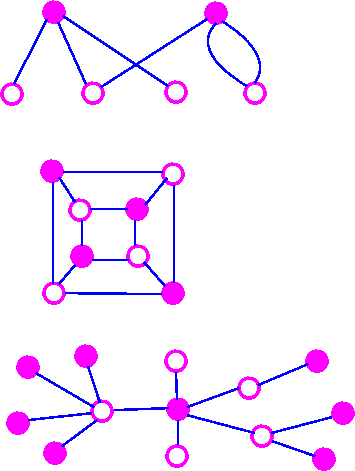


Note that path graph, P*n*, has *n*-1 edges, and can be obtained from cycle graph, C*n*, by removing any edge.

**Bipartite Graphs**

A bipartite graph is a graph whose vertex-set can be split into two sets in such a way that each edge of the graph joins a vertex in first set to a vertex in second set.

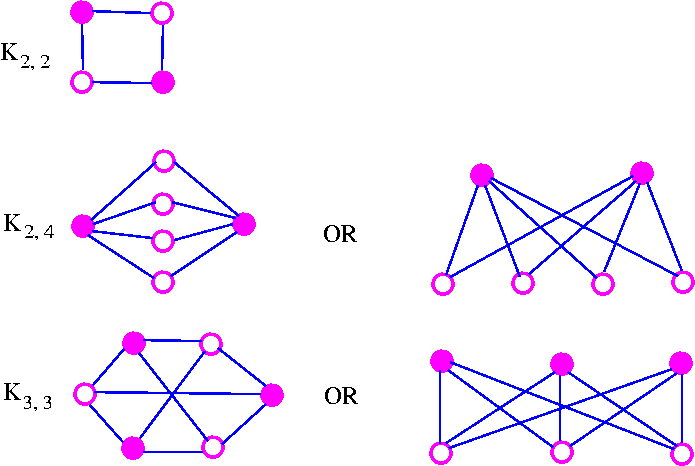
The examples of bipartite graphs are:



**Complete Bipartite Graph**

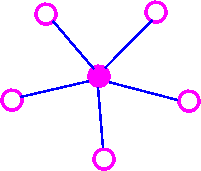
A complete bipartite graph is a bipartite graph in which each vertex in the first set is joined to each vertex in the second set  by exactly one edge. The complete bipartite graph with *r* vertices and 3 vertices is denoted by K*r,s*.

The following are some examples.



Note that K*r,s* has r+s vertices (r vertices of degrees, and s vertices of degree r), and rs edges. Note also that  K*r,s*= K*s,r*.

**An Important Note:**    A complete bipartite graph of the form K*r,s* is called a star graph.

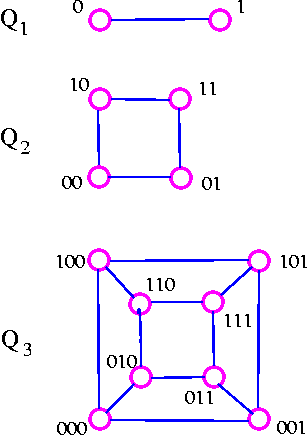


**Cube Graph**

The cube graphs is a bipartite graphs and have appropriate in the coding theory.

The cube graphs constructed by taking as vertices all binary words of a given length and joining two of these vertices if the corresponding binary words differ in just one place. The binary words of length *k* is called the *k*-cube (or *k*-dimensional cube) graph and is denoted by Q*k*.

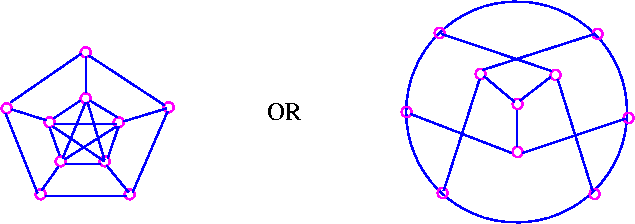
The following are some examples.



Note that Q*k* has 2*k* vertices and is regular of degree *k*. It follows from consequence 3 of the handshaking lemma that Q*k*has *k\** 2*k*-1edges.

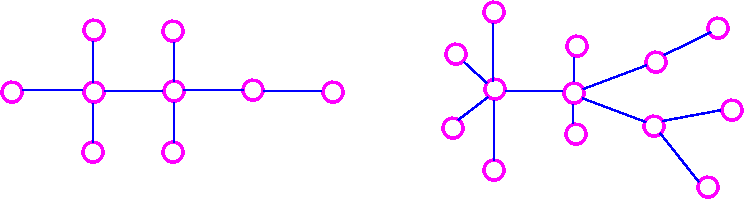
**The Peterson Graph**

This graph is named after a Danish mathematician, Julius Peterson(1839-1910), who discovered the graph in a paper of 1898.



**Tree Graph**

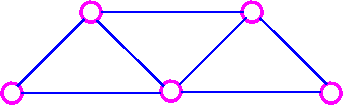
A tree is a connected graph which has no cycles.



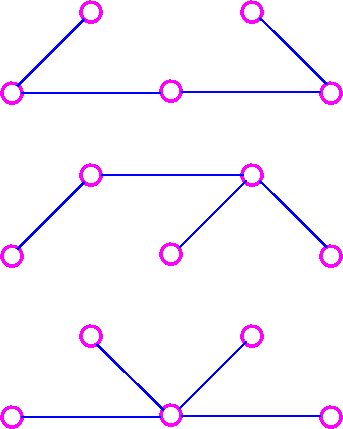
**Spanning Tree**

If *G* is a connected graph, the spanning tree in *G* is a subgraph of *G* which includes every vertex of *G* and  is also a tree.

Consider the following graph

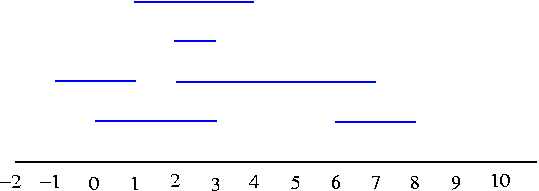


The following are the three of its spanning trees:

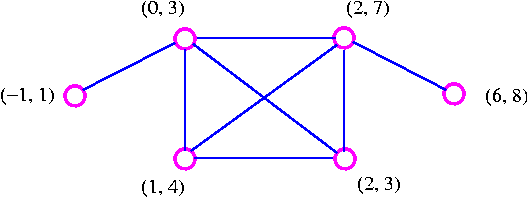


**Interval Graphs**

Consider the intervals (0, 3), (2, 7), (-1, 1), (2, 3), (1, 4), (6, 8) which may be illustrated as



We can construct the resulting interval graphs by taking the interval as vertices, join two of these vertices by an edge whenever the corresponding intervals have at least one point in common.



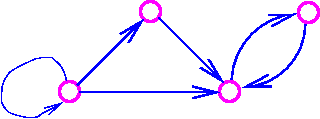
Note that since the intervals (-1, 1) and (1, 4) are open intervals, they do not have a point in common.

**Definition of Diagraph**

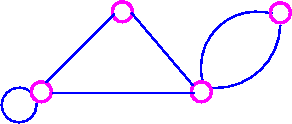
A directed graph or diagraph *D* consists of a set of elements, called vertices, and a list of ordered pairs of these elements, called arcs. The set of vertices is called arcs. The set of vertices is called the vertex-set of *D*, denoted by *V*(*D*), and the list of arcs is called the arc-list of*D*, denoted by *A*(*D*). If v and w are vertices of *D*, then an arc of the form *vw* is said to be directed from v to *w*, or to join *v* to *w*.

C:\Documents and Settings\admin\Desktop\defEx_files\g34.gif

 The underlying graph of diagraph is the graph obtained by replacing each arc of diagraph by corresponding (undirected) edge. For example, consider the following digraph

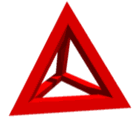


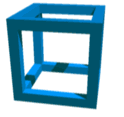
The underlying graph of the above digraph is



**The Platonic Graphs**

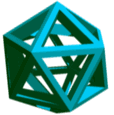
The following regular solids are called the Platonic solids:

  
Tetrahedron

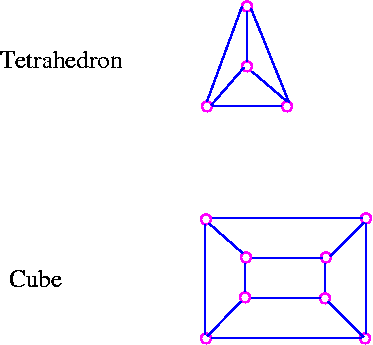
  
Hexahedron (cube)

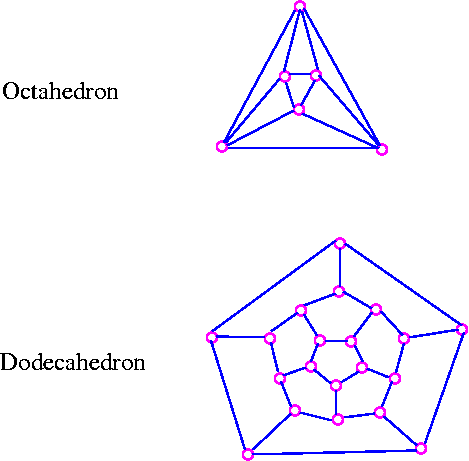
  
Octahedron

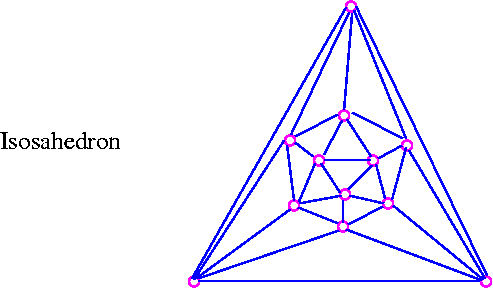
  
Dodecahedron

  
Icosahedron

The name Platonic arises from the fact that these five solids were mentioned in Plato's Timaeus. A Platonic graph is obtained by projecting the corresponding solid on to a plane.







C:\Documents and Settings\admin\Desktop\defEx_files\redline.gif