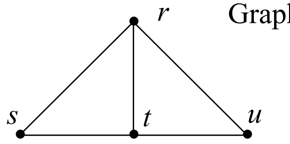
GRAPH THEORY Graphs are used widely for describing Data Structures and networks.

Two non-pictorial descriptions of graphs

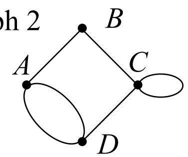
(a) The vertex and edge list description

The most obvious way to describe a graph without drawing a picture is to list the elements of the vertex set and the elements of the edge set. For example Graph 1 and Graph 2 drawn below can be described by the lists underneath each graph.

Graph 1



Graph 2



$V_1 = \{r, s, t, u\}$ $V_2 = \{A, B, C, D\}$

$E_1 = \{rs, rt, ru, st, tu\}$ $E_2 = \{AB, AD, AD, BC, CC, CD\}$

In such a description the order of the vertices and of the edges is irrelevant (as is the order in which each pair of vertices is listed to describe a given edge). However, it does matter whether or not an edge is listed more than once – when this is done it describes a set of multiple edges.

(b) The adjacency matrix description

Another way of storing information about the number of vertices and the number of edges joining each pair is to use an $n \times n$ matrix to describe a graph with n vertices: the element in the i^{th} row and j^{th} column gives the number of edges connecting the i^{th} and j^{th} vertices. For example, the matrices below are the adjacency matrices for Graph 1 and Graph 2 drawn above.

Adjacency matrix for Graph 1 Adjacency matrix for Graph 2

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 0 \end{bmatrix}$$

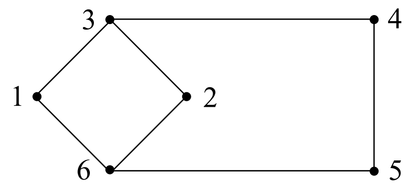
Notice that each matrix is symmetrical about the leading diagonal (why?) and that the absence of loops results in zeros down the diagonal for Graph 1. The matrix describing the graph is not unique: it depends upon the order in which the vertices are listed. The matrix above corresponded to the vertex order r, s, t, u; as an exercise you could try calculating the matrix given by a different order such as s, u, r, t. With some thought you can actually work out which order the vertices are used to produce the above adjacency matrix for Graph 2.

ISOMORPHISMS

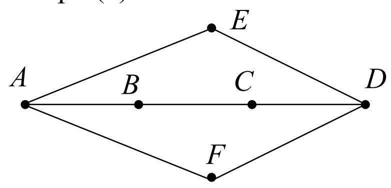
Definition Two graphs G and H are said to be **isomorphic** if there is a one-to-one function f from the vertex set of G onto the vertex set of H such that, for every pair of vertices a, b in G, the number of edges joining a and b in G is the same as the number of edges joining $f(a)$ and $f(b)$ in H. A function f with this property is called an **isomorphism** from G onto H.

Example 1

Graph (a)



Graph (b)



There are 4 possible isomorphisms given by:

Isomorphism 1 $1 \to E$ $2 \to F$ $3 \to A$ $4 \to B$ $5 \to C$ $6 \to D$

Isomorphism 2 $1 \to E$ $2 \to F$ $3 \to D$ $4 \to C$ $5 \to B$ $6 \to A$

Isomorphism 3 $1 \to F$ $2 \to E$ $3 \to A$ $4 \to B$ $5 \to C$ $6 \to D$

Isomorphism 4 $1 \to F$ $2 \to E$ $3 \to D$ $4 \to C$ $5 \to B$ $6 \to A$

An alterative description for an isomorphism is using a function f.

Isomorphism 2 $f(1) = E$ $f(2) = F$ $f(3) = D$

$f(4) = C$ $f(5) = B$ $f(6) = A$

How many pairs of vertices must be considered to check whether or not two graphs are isomorphic ?

. . . .

The examples above illustrate how to prove that two graphs are isomorphic: to do so we must define a suitable isomorphism f. Although in simple cases it is obvious from a picture, we should also check that the number of edges joining *each* pair of vertices in the first graph is the same as the number of edges joining the corresponding pair of vertices in the second graph. It is not as easy to find a method for proving that graphs are not isomorphic: to prove that we must show that no possible choice of function f can give an isomorphism.

Definition The **degree** of a vertex is the number of edges ending at that vertex (with the convention that each loop is counted twice when calculating the degree).

Definition Two vertices are **adjacent** if they are connected (directly) by an edge

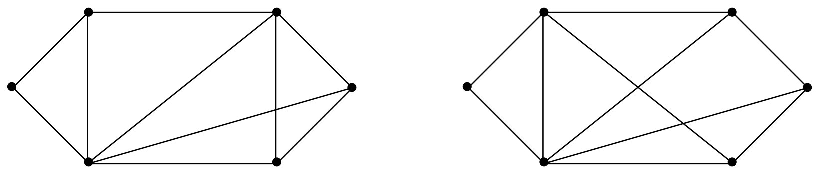
The type of graph properties that are preserved by **isomorphisms** are **degree** and **adjacency**.

Showing Two Graphs are not Isomorphic

Three useful ways of distinguishing graphs from each other are:

1 " `adjacency and degree`" **2** "`circuits`" and **3** "`planarity`".

Example 2 Begin by writing the degree beside each vertex



A B

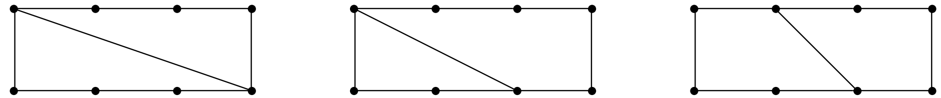
To distinguish these two graphs we note that in Graph A the vertex of degree 2 is adjacent to vertices of degree 3 & degree 5. In Graph B the vertex of degree 2 is adjacent to vertices of degree 4 & degree 5. Alternatively we note that in Graph B there is a vertex of degree 3 which is adjacent to 2 vertices of degree 3. There is no such vertex in Graph A.

Definition A **path** is a finite sequence of distinct edges of the form: $\{v_1v_2, v_2v_3, v_3v_4, \ldots, v_{n-1}v_n\}$

Often abbreviated to: $v_1v_2v_3\cdots v_{n-1}v_n$. The first description is the path described by a sequence of edges. The second description is the path described by a sequence of vertices.

Definition A **circuit** is a path with $v_1 = v_n$.

Example 4 Graphs C and D are not isomorphic by reason of circuits. Graphs C and E are in fact isomorphic.



C D E

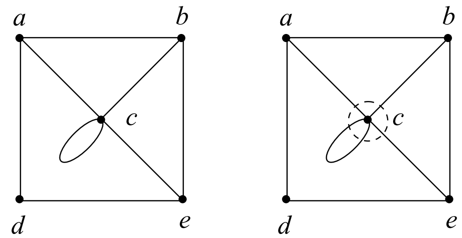
Graph D has a circuit with ... edges, Graph C does not.

Three Methods for Calculating the Degree of a Vertex

1. Draw a small circle around the vertex and count the number of lines.

The degree of vertex c is ....

The degree of vertex e is ....



2. Sum the numbers (counting the diagonal entry twice) in the corresponding row in the adjacency matrix.

Degree of $c = \sum_{x \in V} (E_{cx}) + E_{cc}$ where E_{cx} is the number of edges joining vertex c to vertex x.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

3. In the Edge-Vertex List description, simply sum the number of occurences of the vertex in the Edge List.

In the graph above, $V = \{a, b, c, d, e\}$ $E = \{ab, ac, ad, bc, be, cc, ce, de\}$

The degree of c is 5, the degree of e is 3.