## UNIT - 3

### PERMUTATIONS

Any arrangement of a set of $n$ objects in a given order is called a *permutation* of the object (taken all at a time). Any arrangement of any $r$ $n$ of these objects in a given order is called an "$r$-permutation" or "a permutation of the $n$ objects taken $r$ at a time." Consider, for example, the set of letters $A$, $B$, $C$, $D$. Then:

(i) *BDCA*, *DCBA*, and *ACDB* are permutations of the four letters (taken all at a time).
(ii) *BAD*, *ACB*, *DBC* are permutations of the four letters taken three at a time.
(iii) *AD*, *BC*, *CA* are permutations of the four letters taken two at a time.

We usually are interested in the number of such permutations without listing them. The number of permutations of $n$ objects taken $r$ at a time will be denoted by

$P(n, r)$ (other texts may use ${}_nP_r$, $P_{n,r}$, or $(n)_r$).

The following theorem applies.

**Theorem 5.4:**

$$P(n,r) = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

We emphasize that there are $r$ factors in $n(n-1)(n-2)\cdots(n-r+1)$.

**EXAMPLE** Find the number $m$ of permutations of six objects, say, $A$, $B$, $C$, $D$, $E$, $F$, taken three at a time. In other words, find the number of "three-letter words" using only the given six letters without repetition.

Let us represent the general three-letter word by the following three positions:

____, ____, ____

The first letter can be chosen in 6 ways; following this the second letter can be chosen in 5 ways; and, finally, the third letter can be chosen in 4 ways. Write each number in its appropriate position as follows:

$\underline{6}$, $\underline{5}$, $\underline{4}$

By the Product Rule there are $m$ 6 5 4 120 possible three-letter words without repetition from the six letters. Namely, there are 120 permutations of 6 objects taken 3 at a time. This agrees with the formula in Theorem 5.4:

$$P(6, 3) = 6 \cdot 5 \cdot 4 = 120$$

In fact, Theorem 5.4 is proven in the same way as we did for this particular case.

Consider now the special case of $P(n, r)$ when $r = n$. We get the following result.

**Corollary :** There are $n!$ permutations of $n$ objects (taken all at a time).

For example, there are $3! = 6$ permutations of the three letters $A$, $B$, $C$. These are:

*ABC*, *ACB*, *BAC*, *BCA*, *CAB*, *CBA*.

## Permutations with Repetitions

Frequently we want to know the number of permutations of a multiset, that is, a set of objects some of which are alike. We will let

$$P(n;\, n_1, n_2, \ldots, n_r)$$

denote the number of permutations of $n$ objects of which $n_1$ are alike, $n_2$ are alike, .. ., $n_r$ are alike. The general formula follows:

**Theorem 5.6:** $P(n;\, n_1, n_2, \ldots, n_r) = \dfrac{n!}{n_1!\,n_2!\ldots n_r!}$

We indicate the proof of the above theorem by a particular example. Suppose we want to form all possible five-letter "words" using the letters from the word "*BABBY*." Now there are $5! = 120$ permutations of the objects $B_1, A, B_2, B_3, Y$, where the three $B$'s are distinguished. Observe that the following six permutations

$$B_1B_2B_3AY,\ B_2B_1B_3AY,\ B_3B_1B_2AY,\ B_1B_3B_2AY,\ B_2B_3B_1AY,\ B_3B_2B_1AY$$

produce the same word when the subscripts are removed. The 6 comes from the fact that there are $3! = 3\cdot2\cdot1 = 6$ different ways of placing the three $B$'s in the first three positions in the permutation. This is true for each set of three positions in which the $B$'s can appear. Accordingly, the number of different five-letter words that can be formed using the letters from the word "*BABBY*" is:

$$P(5;\,3) = \frac{5!}{3!} = 20$$

**EXAMPLE:** Find the number $m$ of seven-letter words that can be formed using the letters of the word "*BENZENE*."
We seek the number of permutations of 7 objects of which 3 are alike (the three $E$'s), and 2 are alike (the two $N$'s). By Theorem 5.6,

$$m = P(7;\,3,\,2) = \frac{7!}{3!2!} = \frac{7\cdot6\cdot5\cdot4\cdot3\cdot2\cdot1}{3\cdot2\cdot1\cdot2\cdot1} = 420$$

## Ordered Samples

Many problems are concerned with choosing an element from a set $S$, say, with $n$ elements. When we choose one element after another, say, $r$ times, we call the choice an *ordered sample* of size $r$. We consider two cases.

### (1) Sampling with replacement

Here the element is replaced in the set $S$ before the next element is chosen. Thus, each time there are $n$ ways to choose an element (repetitions are allowed). The Product rule tells us that the number of such samples is:

$$n \cdot n \cdot n \cdots n \cdot n(\text{r factors}) = n^r$$

**(2) Sampling without replacement**
Here the element is not replaced in the set *S* before the next element is chosen. Thus, there is no repetition in the ordered sample. Such a sample is simply an *r*-permutation. Thus the number of such samples is:

$$P(n, r) = n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}$$

**EXAMPLE:** Three cards are chosen one after the other from a 52-card deck. Find the number *m* of ways this can be done: (a) with replacement; (b) without replacement.

(a) Each card can be chosen in 52 ways. Thus m = 52(52)(52) = 140 608.
(b) Here there is no replacement. Thus the first card can be chosen in 52 ways, the second in 51 ways, and the third in 50 ways. Therefore:

m = P(52, 3) = 52(51)(50) = 132 600

**COMBINATIONS**
Let *S* be a set with *n* elements. A *combination* of these *n* elements taken *r* at a time is any selection of *r* of the elements where order does not count. Such a selection is called an *r-combination*; it is simply a subset of *S* with *r* elements. The number of such combinations will be denoted by

$C(n, r)$ (other texts may use ${}_nC_r$, $C_{n,r}$, or $C^n$).

Before we give the general formula for $C(n, r)$, we consider a special case.

**EXAMPLE 5.7** Find the number of combinations of 4 objects, *A*, *B*, *C*, *D*, taken 3 at a time. Each combination of three objects determines 3!= 6 permutations of the objects as follows:

*ABC* : *ABC*, *ACB*, *BAC*, *BCA*, *CAB*,
*CBA* *ABD* : *ABD*, *ADB*, *BAD*, *BDA*,
*DAB*, *DBA* *ACD* : *ACD*, *ADC*, *CAD*,
*CDA*, *DAC*, *DCA* *BCD* : *BDC*, *BDC*,
*CBD*, *CDB*, *DBC*, *DCB*

Thus the number of combinations multiplied by 3! gives us the number of permutations; that is,

$$C(4, 3) \cdot 3! = P(4, 3) \quad \text{or} \quad C(4, 3) = \frac{P(4, 3)}{3!}$$

But $P(4, 3) = 4 \cdot 3 \cdot 2 = 24$ and 3!= 6; hence $C(4, 3) = 4$ as noted above.
As indicated above, any combination of *n* objects taken *r* at a time determines *r*! permutations of the objects in the combination; that is,

$$P(n, r) = r!\, C(n, r)$$

Accordingly, we obtain the following formula for $C(n, r)$ which we formally state as a theorem.

**Theorem 5.7:** $C(n, r) = \dfrac{P(n, r)}{r!} = \dfrac{n!}{r!(n-r)!}$

Recall that the binomial coefficient $\binom{n}{r}$ was defined to be $\dfrac{n!}{r!(n-r)!}$; hence

$$\boxed{C(r, n) = \binom{n}{r}}$$

We shall use $C(n, r)$ and $\binom{n}{r}$ interchangeably.

**EXAMPLE:** A farmer buys 3 cows, 2 pigs, and 4 hens from a man who has 6 cows, 5 pigs, and 8 hens. Find the number *m* of choices that the farmer has.
The farmer can choose the cows in *C(*6, 3*)* ways, the pigs in *C(*5, 2*)* ways, and the hens in *C(*8, 4*)* ways. Thus the number *m* of choices follows:

$$m = \binom{6}{3}\binom{5}{2}\binom{8}{4} = \frac{6 \cdot 5 \cdot 4}{3 \cdot 2 \cdot 1} \cdot \frac{5 \cdot 4}{2 \cdot 1} \cdot \frac{8 \cdot 7 \cdot 6 \cdot 5}{4 \cdot 3 \cdot 2 \cdot 1} = 20 \cdot 10 \cdot 70 = 14\,000$$

**THE PIGEONHOLE PRINCIPLE**

Many results in combinational theory come from the following almost obvious statement.

**Pigeonhole Principle:** If *n* pigeonholes are occupied by *n* + 1 or more pigeons, then at least one pigeonhole is occupied by more than one pigeon.
This principle can be applied to many problems where we want to show that a given situation can occur.
**EXAMPLE:**
(a) Suppose a department contains 13 professors, then two of the professors (pigeons) were born in the same month (pigeonholes).

(b) Find the minimum number of elements that one needs to take from the set *S* = {1, 2, 3, . . . , 9} to be sure that two of the numbers add up to 10.
Here the pigeonholes are the five sets {1, 9}, {2, 8}, {3, 7}, {4, 6}, {5}. Thus any choice of six elements (pigeons) of *S* will guarantee that two of the numbers add up to ten.

The Pigeonhole Principle is generalized as follows.

**Generalized Pigeonhole Principle:** If *n* pigeonholes are occupied by *kn* + 1 or more pigeons, where *k* is a positive integer, then at least one pigeonhole is occupied by *k* + 1 or more pigeons.

**EXAMPLE:** Find the minimum number of students in a class to be sure that three of them are born in the same month.

Here the $n = 12$ months are the pigeonholes, and $k + 1 = 3$ so $k = 2$. Hence among any $kn + 1 = 25$ students (pigeons), three of them are born in the same month.

# UNIT-2

## DIRECTED GRAPHS

A *directed graph G* or *digraph* (or simply *graph*) consists of two things:
(i) A set $V$ whose elements are called *vertices*, *nodes*, or *points*.
(ii) A set $E$ of *ordered* pairs $(u, v)$ of vertices called *arcs* or *directed edges* or simply *edges*.
We will write $G(V, E)$ when we want to emphasize the two parts of $G$. We will also write $V(G)$ and $E(G)$ to denote, respectively, the set of vertices and the set of edges of a graph $G$. (If it is not explicitly stated, the context usually determines whether or not a graph $G$ is a directed graph.)
Suppose $e = (u, v)$ is a directed edge in a digraph $G$. Then the following terminology is used:
(a) $e$ *begins* at $u$ and *ends* at $v$.
(b) $u$ is the *origin* or *initial point* of $e$, and $v$ is the *destination* or *terminal point* of $e$.
(c) $v$ is a *successor* of $u$.
(d) $u$ is *adjacent to* $v$, and $v$ is *adjacent from* $u$.
If $u = v$, then $e$ is called a *loop*.
The set of all successors of a vertex $u$ is important; it is denoted and formally defined by

$$\text{succ}(u) = \{v \in V \mid \text{there exists an edge } (u, v) \in E\}.$$

It is called the *successor list* or *adjacency list* of $u$.

A *picture* of a directed graph $G$ is a representation of $G$ in the plane. That is, each vertex $u$ of $G$ is represented by a dot (or small circle), and each (directed) edge $e = (u, v)$ is represented by an arrow or directed curve from the initial point $u$ of $e$ to the terminal point $v$. One usually presents a digraph $G$ by its picture rather than explicitly listing its vertices and edges.
If the edges and/or vertices of a directed graph $G$ are labeled with some type of data, then $G$ is called a *labeled directed graph*.
A directed graph $(V, E)$ is said to be *finite* if its set $V$ of vertices and its set $E$ of edges are finite.

**EXAMPLE 1**

(a) Consider the directed graph $G$ pictured in Fig. 9-1($a$). It consists of four vertices, $A$, $B$, $C$, $D$, that is, $V(G) = \{A,B,C,D\}$ and the seven following edges:

$$E(G) = \{e1, e2, \dots, e7\} = \{(A,D), (B, A), (B, A), (D,B), (B,C), (D,C), (B, B)\}$$

The edges $e2$ and $e3$ are said to be *parallel* since they both begin at $B$ and end at $A$. The edge $e7$ is a *loop* since it begins and ends at $B$.

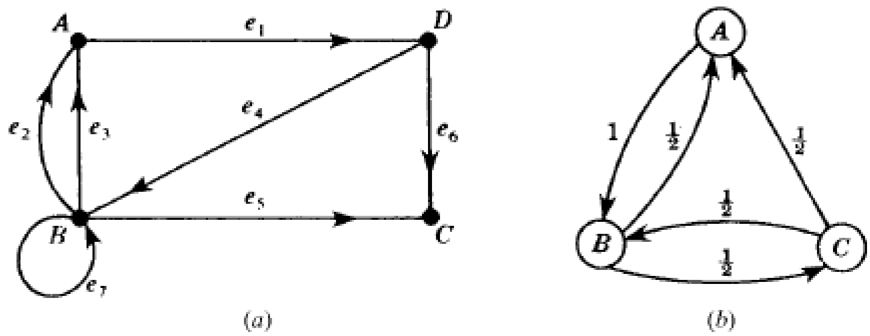



**Fig. 9-1**

(b) Suppose three boys, A,B,C, are throwing a ball to each other such that A always throws the ball to B, but B and C are just as likely to throw the ball to A as they are to each other. This dynamic system is pictured in Fig. 9-1(b) where edges are labeled with the respective probabilities, that is, A throws the ball to B with probability 1, B throws the ball to A and C each with probability 1/2, and C throws the ball to A and B each with probability 1/2.

**Subgraphs**

Let $G = G(V,E)$ be a directed graph, and let $V`$ be a subset of the set $V$ of vertices of $G$. Suppose $E`$ is a subset of $E$ such that the endpoints of the edges in $E`$ belong to $V`$. Then $H(V`,E`)$ is a directed graph, and it is called a *subgraph* of $G$. In particular, if $E`$ contains all the edges in $E$ whose endpoints belong to $V`$, then $H(V`,E`)$ is called the subgraph of $G$ *generated* or *determined* by $V`$. For example, for the graph $G = G(V,E)$ in Fig. 9-1(*a*), $H(V`,E`)$ is the subgraph of $G$ determine by the vertex set $V`$ where

$$V' = \{B, C, D\} \text{ and } E`=(e4, e5, e6, e7)= \{(D, B), (B, C), (D, C), (B, B)\}$$

**BASIC DEFINITIONS**

This section discusses the questions of degrees of vertices, paths, and connectivity in a directed graph.

**Degrees**

Suppose $G$ is a directed graph. The *outdegree* of a vertex $v$ of $G$, written outdeg($v$), is the number of edges beginning at $v$, and the *indegree* of $v$, written indeg($v$), is the number of edges ending at $v$. Since each edge begins and ends at a vertex we immediately obtain the following theorem.

**Theorem 9.1:** The sum of the outdegrees of the vertices of a digraph $G$ equals the sum of the indegrees of the vertices, which equals the number of edges in $G$.

A vertex $v$ with zero indegree is called a *source*, and a vertex $v$ with zero outdegree is called a *sink*.

**EXAMPLE 2** Consider the graph $G$ in Fig. 9-1(*a*).We have:

$$\text{outdeg } (A) = 1, \text{ outdeg } (B)= 4, \text{ outdeg } (C) = 0, \text{ outdeg } (D)= 2,$$
$$\text{indeg } (A) = 2, \text{ indeg } (B)= 2, \text{ indeg } (C) = 2, \text{ indeg } (D) = 1.$$

As expected, the sum of the outdegrees equals the sum of the indegrees, which equals the number 7 of edges.The vertex $C$ is a sink since no edge begins at $C$. The graph has no sources.

## Paths

Let $G$ be a directed graph. The concepts of path, simple path, trail, and cycle carry over from nondirected graphs to the directed graph $G$ except that the directions of the edges must agree with the direction of the path. Specifically:

(i) A (*directed*) *path* $P$ in $G$ is an alternating sequence of vertices and directed edges, say,

$$P = (v_0, e_1, v_1, e_2, v_2, \ldots, e_n, v_n)$$

such that each edge $ei$ begins at $v_i{-}1$ and ends at $v_i$ . If there is no ambiguity, we denote $P$ by its sequence of vertices or its sequence of edges.

(ii) The *length* of the path $P$ is $n$, its number of edges.

(iii) A *simple path* is a path with distinct vertices. A *trail* is a path with distinct edges.

(iv) A *closed path* has the same first and last vertices.

(v) A *spanning path* contains all the vertices of $G$.

(vi) A *cycle* (or *circuit*) is a closed path with distinct vertices (except the first and last).

(vii) A*semipath* is the same as a path except the edge $ei$ may begin at $vi{-}1$ or $vi$ and end at the other vertex. *Semitrails* and *semisimple paths* are analogously defined.

A vertex $v$ is *reachable* from a vertex $u$ if there is a path from $u$ to $v$. If $v$ is reachable from $u$, then (by eliminating redundant edges) there must be a simple path from $u$ to $v$.

**EXAMPLE 9.3** Consider the graph $G$ in Fig. 9-1($a$).

(a) The sequence $P_1 = (D,C,B,A)$ is a semipath but not a path since $(C,B)$ is not an edge; that is, the direction of $e_5 = (C,B)$ does not agree with the direction of $P1$.

(b) The sequence $P_2 = (D,B,A)$ is a path from $D$ to $A$ since $(D, B)$ and $(B, A)$ are edges. Thus $A$ is reachable from $D$.

## Connectivity

There are three types of connectivity in a directed graph $G$:

(i) $G$ is *strongly connected* or *strong* if, for any pair of vertices $u$ and $v$ in $G$, there is a path from $u$ to $v$ and a path from $v$ to $u$, that is, each is reachable from the other.

(ii) $G$ is *unilaterally connected* or *unilateral* if, for any pair of vertices $u$ and $v$ in $G$, there is a path from $u$ to $v$ or a path from $v$ to $u$, that is, one of them is reachable from the other.

(iii) $G$ is *weakly connected* or *weak* if there is a semipath between any pair of vertices $u$ and $v$ in $G$.

Let $G`$ be the (nondirected) graph obtained from a directed graph $G$ by allowing all edges in $G$ to be nondirected. Clearly, $G$ is weakly connected if and only if the graph $G`$ is connected.

Observe that strongly connected implies unilaterally connected which implies weakly connected. We say that $G$ is *strictly unilateral* if it is unilateral but not strong, and we say that $G$ is *strictly weak* if it is weak but not unilateral.

Connectivity can be characterized in terms of spanning paths as follows:

**Theorem 9.2:** Let $G$ be a finite directed graph. Then:
(i) $G$ is strong if and only if $G$ has a closed spanning path.
(ii) $G$ is unilateral if and only if $G$ has a spanning path.
(iii) $G$ is weak if and only if $G$ has a spanning semipath.

**EXAMPLE:** Consider the graph $G$ in Fig. 9-1(*a*). It is weakly connected since the underlying nondirected graph is connected. There is no path from $C$ to any other vertex, that is, $C$ is a sink, so $G$ is not strongly connected. However, $P = (B,A,D,C)$ is a spanning path, so $G$ is unilaterally connected.
Graphs with sources and sinks appear in many applications (such as flow diagrams and networks).A sufficient condition for such vertices to exist follows.

**Theorem 9.3:** Suppose a finite directed graph $G$ is cycle-free, that is, contains no (directed) cycles. Then $G$ contains a source and a sink.
**Proof:** Let $P = (v_0, v_1, \dots, v_n)$ be a simple path of maximum length, which exists since $G$ is finite. Then the last vertex $v_n$ is a sink; otherwise an edge $(v_n, u)$ will either extend $P$ or form a cycle if $u = v_i$ , for some $i$. Similarly, the first vertex $v_0$ is a source.

## SEQUENTIAL REPRESENTATION OF DIRECTED GRAPHS

There are two main ways of maintaining a directed graph $G$ in the memory of a computer. One way, called the *sequential representation* of $G$, is by means of its adjacency matrix $A$. The other way, called the *linked representation* of $G$, is by means of linked lists of neighbors. This section covers the first representation. The linked representation will be covered in Section 9.7.
Suppose a graph $G$ has $m$ vertices (nodes) and $n$ edges. We say $G$ is *dense* if $m = O(n2)$ and *sparse* if $m = O(n)$ or even if $m = O(n \log n)$. The matrix representation of $G$ is usually used when $G$ is dense, and linked lists are usually used when $G$ is sparse. Regardless of the way one maintains a graph $G$ in memory, the graph $G$ is normally input into the computer by its formal definition, that is, as a collection of vertices and a collection of edges (pairs of vertices).

**Remark:** In order to avoid special cases of our results, we assume, unless otherwise stated, that $m>1$ where $m$ is the number of vertices in our graph $G$. Therefore, $G$ is not connected if $G$ has no edges.

### Digraphs and Relations, Adjacency Matrix

Let $G(V, E)$ be a *simple* directed graph, that is, a graph without parallel edges. Then $E$ is simply a subset of $V \times V$, and hence $E$ is a relation on $V$ . Conversely, if $R$ is a relation on a set $V$, then $G(V, R)$ is a simple directed graph. Thus the concepts of relations on a set and simple directed graphs are one and the same.
Suppose $G$ is a simple directed graph with $m$ vertices, and suppose the vertices of $G$ have been ordered and are called $v_1, v_2,\dots,v_m$. Then the *adjacency matrix* $A = [a_{ij}]$ of $G$ is the $m \times m$ matrix defined as follows:

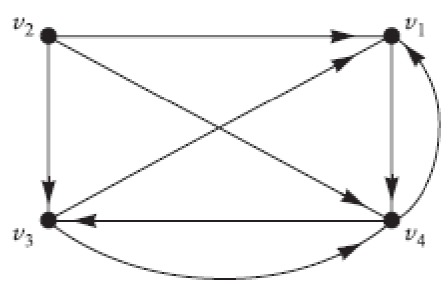
$$a_{ij} = \begin{cases} 1 & \text{if there is an edge } (v_i, v_j) \\ 0 & \text{otherwise} \end{cases}$$

Such a matrix $A$, which contains entries of only 0 or 1, is called a *bit matrix* or a *Boolean matrix*. (Although the adjacency matrix of an undirected graph is symmetric, this is not true here for a directed graph.)
The adjacency matrix $A$ of the graph $G$ does depend on the ordering of the vertices of $G$. However, the matrices resulting from two different orderings are closely related in that one can be obtained from the other by simply interchanging rows and columns. Unless otherwise stated, we assume that the vertices of our matrix have a fixed ordering.
**Remark:** The adjacency matrix $A = [a_{ij}]$ may be extended to directed graphs with parallel edges by setting:

$$a_{ij} = \text{the number of edges beginning at } v_i \text{ and ending at } v_j$$

Then the entries of $A$ will be nonnegative integers. Conversely, every $m \times m$ matrix $A$ with nonnegative integer entries uniquely defines a directed graph with $m$ vertices.

**EXAMPLE:** Let $G$ be the directed graph in Fig. 9-4($a$) with vertices $v_1$, $v_2$, $v_3$, $v_4$. Then the adjacency matrix $A$ of $G$ appears in Fig. 9-4($b$). Note that the number of 1's in $A$ is equal to the number (eight) of edges.




($a$)

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

($b$)

**Fig. 9-4**

Consider the powers $A$, $A2$, $A3$,… of the adjacency matrix $A = [a_{ij}]$ of a graph $G$. Let

$$a_k(i, j) = \text{the } ij \text{ entry in the matrix } A^k$$

Note that $a_1(i, j) = aij$ gives the number of paths of length 1 from vertex $vi$ to vertex $vj$ . One can show that $a_2(i, j)$ gives the number of paths of length 2 from $v_i$ to $v_j$ .

**Proposition 9.4:** Let $A$ be the adjacency matrix of a graph $G$. Then $a_k(i, j)$, the $ij$ entry in the matrix $A^k$, gives the number of paths of length $K$ from $v_i$ to $v_j$ .
**EXAMPLE 9.7** Consider again the graph $G$ and its adjacency matrix $A$ appearing in Fig. 9-4. The powers $A^2$, $A^3$, and $A^4$ of $A$ follow:

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad A^3 = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 3 & 0 & 2 & 3 \\ 2 & 0 & 1 & 2 \\ 2 & 0 & 2 & 1 \end{bmatrix}, \quad A^4 = \begin{bmatrix} 2 & 0 & 2 & 1 \\ 5 & 0 & 3 & 5 \\ 3 & 0 & 2 & 3 \\ 3 & 0 & 1 & 4 \end{bmatrix}$$

Observe that $a_2(4, 1) = 1$, so there is a path of length 2 from $v4$ to $v1$. Also, $a_3(2, 3) = 2$, so there are two paths of length 3 from $v2$ to $v3$; and $a_4(2, 4) = 5$, so there are five paths of length 4 from $v_2$ to $v_4$.

**Remark:** Let $A$ be the adjacency matrix of a graph $G$, and let $Br$ be the matrix defined by:

$$Br = A + A^2 + A^3 + \cdots + A^r$$

Then the $ij$ entry of the matrix $Br$ gives the number of paths of length $r$ or less from vertex $v_i$ to vertex $v_j$ .

**Path Matrix**

Let $G = G(V,E)$ be a simple directed graph with $m$ vertices $v_1, v_2, \ldots, v_m$. The *path matrix* or *reachability matrix* of $G$ is the $m$-square matrix $P = [p_{ij}]$ defined as follows:

$$p_{ij} = \begin{cases} 1 & \text{if there is a path from to } v_i \text{ to } v_j \\ 0 & \text{otherwise} \end{cases}$$

(The path matrix $P$ may be viewed as the transitive closure of the relation $E$ on $V$ .)

Suppose now that there is a path from vertex $v_i$ to vertex $v_j$ in a graph $G$ with $m$ vertices. Then there must be a simple path from $v_i$ to $v_j$ when $v_i \neq v_j$ , or there must be a cycle from $v_i$ to $vj$ when $v_i = v_j$ . Since $G$ has $m$ vertices, such a simple path must have length $m - 1$ or less, or such a cycle must have length $m$ or less. This means that there is a nonzero $ij$ entry in the matrix $B_m$ (defined above) where $A$ is the adjacency matrix of $G$. Accordingly, the path matrix $P$ and $B_m$ have the same nonzero entries. We state this result formally.

**Proposition 9.5:** Let $A$ be the adjacency matrix of a graph $G$ with $m$ vertices. Then the path matrix $P$ and $B_m$ have the same nonzero entries where

$$B_m = A + A^2 + A^3 + \cdots + A^m$$

Recall that a directed graph $G$ is said to be *strongly connected* if, for any pair of vertices $u$ and $v$ in $G$, there is a path from $u$ to $v$ and from $v$ to $u$. Accordingly, $G$ is strongly connected if and only if the path matrix $P$ of $G$ has no zero entries. This fact together with Proposition 9.5 gives the following result.

**Proposition 9.6:** Let $A$ be the adjacency matrix of a graph $G$ with $m$ vertices. Then $G$ is strongly connected if and only if $B_m$ has no zero entries where

$$B_m = A + A^2 + A^3 + \cdots + A^m$$

**EXAMPLE 9.8** Consider the graph $G$ and its adjacency matrix $A$ appearing in Fig. 9-4. Here $G$ has $m = 4$ vertices. Adding the matrix $A$ and matrices $A^2$, $A^3$, $A^4$ in Example 9.7, we obtain the following matrix $B_4$ and also path (reachability) matrix $P$ by replacing the nonzero entries in $B_4$ by 1:

$$B_4 = \begin{bmatrix} 4 & 0 & 3 & 4 \\ 11 & 0 & 7 & 11 \\ 7 & 0 & 4 & 7 \\ 7 & 0 & 4 & 7 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Examining the matrix $B_4$ or $P$, we see zero entries; hence $G$ is not strongly connected. In particular, we see that the vertex $v_2$ is not reachable from any of the other vertices.

**Remark:** The adjacency matrix $A$ and the path matrix $P$ of a graph $G$ may be viewed as logical (Boolean) matrices where 0 represents “false” and 1 represents “true.” Thus the logical operations of $\wedge$ (AND) and $\vee$ (OR) may be applied to the entries of $A$ and $P$ where these operations, used in the next section, are defined in Fig. 9-5.

| $\wedge$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

(*a*) AND.

| $\vee$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

(*b*) OR.

**Fig. 9-5**

**Transitive Closure and the Path Matrix**

Let $R$ be a relation on a finite set $V$ with $m$ elements. As noted above, the relation $R$ may be identified with the simple directed graph $G=G\ (V, R)$. We note that the composition relation $R^2 = R \times R$ consists of all pairs $(u, v)$ such that there is a path of length 2 from $u$ to $v$. Similarly:

$$R^k = \{(u, v) | \text{ there is a path of length} K \text{ from } u \text{ to } v\}.$$

The transitive closure $R^*$ of the relation $R$ on $V$ may now be viewed as the set of ordered pairs $(u, v)$ such that there is a path from $u$ to $v$ in the graph $G$. Furthermore, by the above discussion, we need only look at simple paths of length $m - 1$ or less and cycles of length m or less. Accordingly, we have the following result which characterizes the transitive closure $R^*$ of $R$.

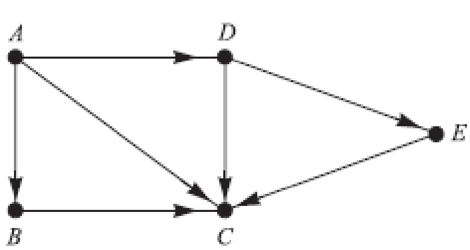
**Theorem 9.7:** Let $R$ be a relation on a set $V$ with $m$ elements. Then:

(i) $R^* = R \cup R^2 \cup \ldots \cup R^m$ is the transitive closure of $R$.

(ii) The path matrix $P$ of $G(V, R)$ is the adjacency matrix of $G`(V, R^*)$.

**LINKED REPRESENTATION OF DIRECTED GRAPHS**

Let $G$ be a directed graph with $m$ vertices. Suppose the number of edges of $G$ is $O(m)$ or even $O(m \log m)$, that is, suppose $G$ is sparse. Then the adjacency matrix $A$ of $G$ will contain many zeros; hence a great deal of memory space will be wasted. Accordingly, when $G$ is sparse, $G$ is usually represented in memory by some type of *linked representation*, also called an *adjacency structure*, which is described below by means of an example.




(*a*) Graph *G*

| Vertex | Adjacency list |
|---|---|
| *A* | *B, C, D* |
| *B* | *C* |
| *C* | ∅ |
| *D* | *C, E* |
| *E* | *C* |

(*b*) Adjacency lists of *G*

**Fig. 9-9**

Consider the directed graph $G$ in Fig. 9-9(*a*). Observe that $G$ may be equivalently defined by the table in Fig. 9-9(*b*), which shows each vertex in $G$ followed by its *adjacency list*, also called its *successors* or *neighbors*. Here the symbol $\varnothing$ denotes an empty list. Observe that each edge of $G$ corresponds to a unique vertex in an adjacency list and vice versa. Here $G$ has seven edges and there are seven vertices in the adjacency lists. This table may also be presented in the following compact form where a colon ":" separates a vertex from its list of neighbors, and a semicolon “;” separates the different lists:

$$G = [A : B,C,D;\ B : C;\ C : \_;\ D : C,E;\ E : C]$$

The *linked representation* of a directed graph *G* maintains *G* in memory by using linked lists for its adjacency lists. Specifically, the linked representation will normally contain two files (sets of records), one called the Vertex File and the other called the Edge File, as follows.

(a) ***Vertex File*:** The Vertex File will contain the list of vertices of the graph *G* usually maintained by an array or by a linked list. Each record of the Vertex File will have the form

| VERTEX | NEXT-V | PTR | |
|---|---|---|---|

Here VERTEX will be the name of the vertex, NEXT-V points to the next vertex in the list of vertices in the Vertex File, and PTR will point to the first element in the adjacency list of the vertex appearing in the Edge File. The shaded area indicates that there may be other information in the record corresponding to the vertex.

(b) ***Edge File*:** The Edge File contains the edges of *G* and also contains all the adjacency lists of *G* where each list is maintained in memory by a linked list. Each record of the Edge File will represent a unique edge in *G* and hence will correspond to a unique vertex in an adjacency list. The record will usually have the form

| EDGE | BEG-V | END-V | NEXT-E | |
|---|---|---|---|---|

Here:

(1) EDGE will be the name of the edge (if it has a name).

(2) BEG-V- points to location in the Vertex File of the initial (beginning) vertex of the edge.

(3) END-V points to the location in theVertex File of the terminal (ending) vertex of the edge. The adjacency lists appear in this field.

(4) NEXT-E points to the location in the Edge File of the next vertex in the adjacency list.

We emphasize that the adjacency lists consist of terminal vertices and hence are maintained by the END-V field. The shaded area indicates that there may be other information in the record corresponding to the edge. We note that the order of the vertices in any adjacency list does depend on the order in which the edges (pairs of vertices) appear in the input.

Figure 9-10 shows how the graph *G* in Fig. 9-9(*a*) may appear in memory. Here the vertices of *G* are maintained in memory by a linked list using the variable START to point to the first vertex. (Alternatively, one could use a linear array for the list of vertices, and then NEXT-V would not be required.) The choice of eight locations for the Vertex File and 10 locations for the Edge File is arbitrary. The additional space in the files will be used if additional vertices or edges are inserted in the graph. Figure 9-10 also shows, with arrows, the adjacency list [*B*, *C*, *D*] of the vertex *A*.

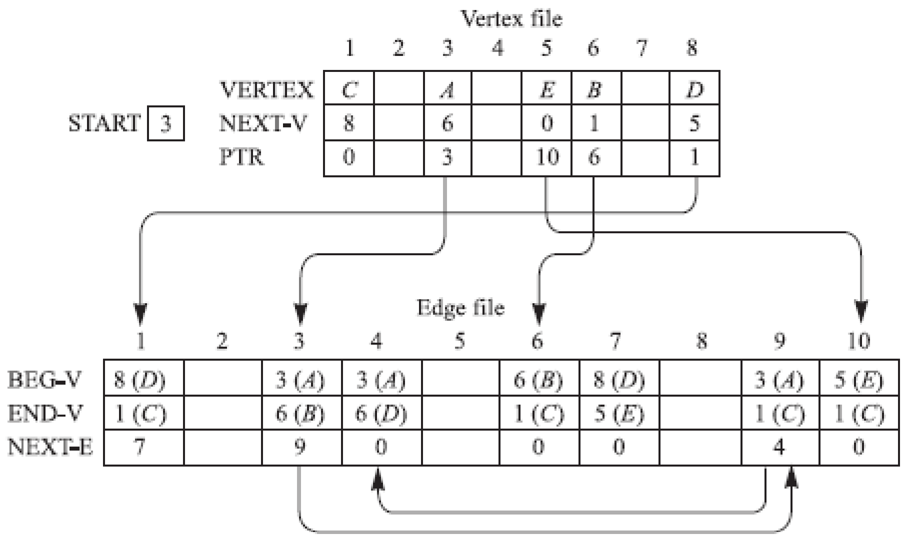



**Fig. 9-10**

## RECURRENCE RELATIONS

### Definition

A recurrence relation is an equation that recursively defines a sequence where the next term is a function of the previous terms (Expressing $F_n$ as some combination of $F_i$ with $i<n$).

**Example** − Fibonacci series $F_n= F_{n-1}+ F_{n-2}$,
Tower of Hanoi $Fn=2F_{n-1}+1$

### Linear Recurrence Relations

A linear recurrence equation of degree k or order k is a recurrence equation which is in the format $x_n= A_1x_{n-1} +A_2 x_{n-1} +A_3 x_{n-1} +……..A_k x_{n-k}$ ($A_n$ is a constant and $A_k\neq 0$) on a sequence of numbers as a first-degree polynomial.

These are some examples of linear recurrence equations −

| Recurrence relations | Initial values | Solutions |
|---|---|---|
| $F_n = F_{n-1} + F_{n-2}$ | $a_1 = a_2 = 1$ | Fibonacci number |
| $F_n = F_{n-1} + F_{n-2}$ | $a_1 = 1, a_2 = 3$ | Lucas Number |
| $F_n = F_{n-2} + F_{n-3}$ | $a_1 = a_2 = a_3 = 1$ | Padovan sequence |
| $F_n = 2F_{n-1} + F_{n-2}$ | $a_1 = 0, a_2 = 1$ | Pell number |

**How to solve linear recurrence relation**

Suppose, a two ordered linear recurrence relation is – $F_n=AF_{n-1}+BF_{n-2}$ where A and B are real numbers.

The characteristic equation for the above recurrence relation is −

$$x^2-Ax-B=0$$

Three cases may occur while finding the roots −

**Case 1** − If this equation factors as $\mathbf{(x-x_1)(x-x_2)=0}$ and it produces two distinct real roots $\mathbf{x_1}$ and $\mathbf{x_2}$, then $\mathbf{F_n=ax_1^n+bx_2^n}$ is the solution. [Here, a and b are constants]
**Case 2** − If this equation factors as $\mathbf{(x-x_1)^2=0}$ and it produces single real root $x_1$, then $\mathbf{F_n=ax_1^n+bnx_1^n}$ is the solution.
**Case 3** − If the equation produces two distinct complex roots, $\mathbf{x_1}$ and $\mathbf{x_2}$ in polar form $\mathbf{x_1=r\angle\theta}$ **and** $\mathbf{x_2=r\angle(-\theta)}$**,** then $\mathbf{F_n=r^n(a\cos(n\theta)+b\sin(n\theta))}$ is the solution.

**Problem 1**

Solve the recurrence relation $F_n=5F_{n-1}-6F_{n-2}$ where $F_0=1$ and $F_1=4$

**Solution**

The characteristic equation of the recurrence relation is −

$$x^2-5x+6=0,$$

So, $(x-3)(x-2)=0$

Hence, the roots are −

$x_1=3$ and $x_2=2$

The roots are real and distinct. So, this is in the form of case 1

Hence, the solution is −

$$F_n=ax_1^n+bx_2^n$$

Here, $F_n=a3^n+b2^n$ (As $x_1=3$ and $x_2=2$)

Therefore,

$1= F_0=a3^0+b2^0 = a+b$
$4= F_1=a3^1+b2^1 =3a+2b$
Solving these two equations, we get $a=2$ and $b=-1$

Hence, the final solution is −

$$F_n=2.3^n+(-1).2^n=2.3^n-2^n$$