

### MATRICES

## SPECIAL MATRICES

→ Only square matrix nxn

- 1. Diagonal Matrix
- 5. Tridiagonal Matrix
- 2. Lower Triangular
- 6. Band Matrix
- 3. Upper Triangular
- Toeplitz Matrix
- 4. Symmetric Matrix 8. Sporse Matrix

### 1. DIAGONAL MATRIX

```
00400
0 0 0 0 6
```

```
ROWS COWMNS ELEMENT TO
                A BE INSERTED
Void set ( int A[], int i, int j, int x)
     if (i==_j)
       A[i-i] = 2;
void get (int A[], int i, int j)
      (نے== نا ¢ن
      return A[i-1];
           return 0;
```

# C++ Class For Diagonal Matrix

```
Class Diagonal ()
                                         void Diagonal " set (int i, int j, int x);
  2
                                                 (ن== ن) F
       private:
                                                    A[i-1] = x;
                 int n;
                int *A;
        public:
                 Diagonal (int n)
                      This → n=n;
                       A = new int(n);
                                               int Diagonal :: get Lint i, int j)
                 3
                                                        if (i==j)
                 void set lint i, int j, int k);
                 void get (inti, int i);
                                                             return 0;
                  void Display();
                  ~ Diagonall)
                       delete []A;
 2
void Diagonal: Display()
                  if (i = = i)
             cout << endl;
```

#### LOWER TRIANGULAR MATRIX

$$M = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

Non Zero = 
$$1+2+3+4+5$$
  
=  $1+2+3+4....n$   
=  $n(n+1)$ 

$$Zero = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$$

## ROW MAJOR

	Q,μ	0.21	022	a <sub>31</sub>	Q <sub>32</sub>	a <sub>33</sub>	ayı	q <sub>42</sub>	aus	a <sub>yy</sub>	ası	$a_{S_2}$	a <sub>53</sub>	asy	۵ <sub>55</sub>
	0	ı	2	3	Ч	5	6	7	8	9	10	17	12	13	14
lmog		<b>₹</b> 0W∑		76W3			TOW 4				γοω 5				

Index 
$$(A[4][3]) = [1+2+3]+2=8$$
  
Index  $(A[5][4]) = [1+2+3+4]+3=13$ 

Index 
$$(A[i][i]) = \left[\frac{i(i-i)}{2}\right] + j-1$$

## COLUMN MAJOR

Index 
$$(A[4][4]) = [5+4+3]+0=12$$
  
Index  $(A[5][4]) = [5+4+3]+1=13$   
Index  $(A[5][3]) = [5+4]+2=11$ 

Index (A[i][i]) = 
$$\begin{bmatrix} n + n-1 + n-2 + ... & n - (i-2) \end{bmatrix} + (i-i)$$
  
=  $\begin{bmatrix} n (i-1) - [1+2+3+... i-2] \end{bmatrix} + (i-i)$   
=  $\begin{bmatrix} n (i-1) - (i-2)(i-1) \end{bmatrix} + (i-i)$ 

### UPPER TRIANGULAR MATRIX

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{21} & a_{23} & a_{24} & a_{25} \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix}$$

$$M[i,j] = 0 \quad \text{if} \quad i \neq j$$

Non Zero = 
$$5+4+3+2+1$$
  
=  $\frac{n(n+1)}{2}$ 

$$Ze_{80} = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$$

# ROW MAJOR

$$INDEX (A[i][i]) = [n+n-1+n-2+...+n-(i-2)] + (j-i)$$

$$= \frac{(i-1)n-(i-2)(i-1)}{2}+(j-i)$$

## COLUMN MAJOR

$$INDEX (A[i][i]) = [1+2+3+....j-1]+i-1 = [i(i-1)]+i-1$$

### SYMMETRIC MATRIX

if 
$$M[i,j] = M[j,i]$$

Either we can store lower triangular matrix, or we can store upper triangular matrix

### TRI DIAGONAL MATRIX

$$M = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix}$$

```
Main diagonal i-j=0

Lower diagonal i-j=1

Upper diagonal i-j=-1
```

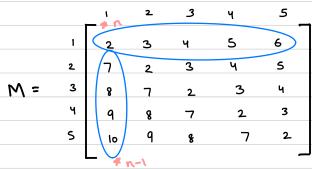
Index (A[i][i])

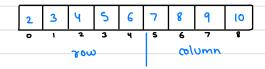
$$M[i,j] = Non Zero if | i-j| \le 1$$
  
 $M[i,j] = Zero if | i-j| > 1$ 

### SQUARE BAND MATRIX (Same as TRI DIAGONAL matrix)

When there are more than one diagonals below the main diagonal and the number of lower and upper diagonal is equal.

## TOEPLITZ MATRIX





$$M[i,j] = M[i-1,j-1]$$

No of elements we want to store: n + n-1

Index A[i](j)

case 1: if 
$$i < = j$$
 Index =  $j-1$ 

### CREATING A DYNAMICALLY ALLOCATED ARRAY

```
int *A,n;

printf (" Enter dimension");

scanf (" ".d", In);

A = (int*) malloc (n* size of (int));

A = new int [n]; (++
```