Sparse Matrices

* **what is a Matrix ?**

A matrix can be defined as a 2D array having 'm' rows and 'n' columns. A matrix with 'm' rows & 'n' columns is called m x n matrix.

It is a set of numbers that are arranged in the horizontal or vertical lines of entries.

For example -

$$\begin{array}{cc} & \text{rows } (m) \rightarrow \\ A = & \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{array} \begin{array}{c} \text{column } (n) \\ \downarrow \end{array}$$

* **what is a Sparse Matrix ?**

Sparse Matrix are those matrices whose majority of their elements is equal to zero.

In other words, the sparse matrix can be defined as the matrix that has greater no. of zero elements than the non-zero elements.

we can also use the simple matrix to store data, then why the sparse matrix is required.

* ~~why is sparse matrix required if we~~ can use the simple matrix to store elements ?

i) Sparse matrix contains lesser non-zero elements than zero, so less memory can be used to store elements. It evaluates only the non-zero elements.

ii) In the case of searching in sparse matrix, we need to traverse only the non-zero elements rather than traversing all the sparse matrix elements. It saves computing time by logically designing a data structure traversing non-zero elements.

iii) Sparse data can easily compressed & thus reduces memory usage. It makes storage of the data more efficiently.

* Types of Sparse Matrix :-

There are 3 types of Sparse Matrices :

i) Lower Triangular Sparse Matrix

ii) Upper Triangular Sparse Matrix.

iii) Tri-Diagonal Matrix.

1 Lower Triangular Sparse Matrix

In this matrix, all elements above the main diagonal have a zero value.

1	0	0	0	0
2	3	0	0	0
5	9	4	0	0
14	17	11	5	0
13	16	19	21	6

all zero elements.

non zero element

all the elements having non-zero value are appear below the diagonal.

A lower triangular matrix arr of size n*n has one non-zero element in the first row, two non-zero elements in the second row & similarly n non-zero elements in the n^{th} row.

We use one-dimensional array to store a lower triangular matrix efficiently in the memory. This array stores only non-zero elements. To store in a one-dimensional array, we do the mapping between a 2D matrix & 1D array. We can done the mapping in any one of the following ways:-

a) Row Wise Mapping
b) Column Wise Mapping

1	0	0	0	0
2	2	0	0	0
1	4	3	0	0
9	8	7	1	0
1	2	7	8	9

if we do row wise mapping of this matrix, then it will be :-

{1, 2, 2, 1, 4, 3, 9, 8, 7, 1, 1, 2, 7, 8, 9}

if we do column wise mapping of this matrix, then it will be :-

{1, 2, 1, 9, 1, 2, 4, 8, 2, 3, 7, 7, 1, 8, 9}

(2) Upper Triangular Sparse Matrix

In this matrix, all elements below the main diagonal have a zero value.

1	2	5	9	7
0	3	10	13	17
0	0	4	21	27
0	0	0	5	30
0	0	0	0	6

all zero elements

all non zero elements

If we do the row wise mapping of the matrix then it will be :-

(5) {1, 2, 5, 9, 7, 3, 10, 13, 17, 4, 21, 27, 5, 30, 6}

If we do the column wise mapping of the matrix then it will be :-

{1, 2, 3, 5, 10, 4, 9, 13, 21, 5, 7, 17, 27, 30, 6}

(3) Tri Diagonal Sparse Matrix

In this matrix, elements with a non-zero value appear only on the diagonal or immediately below or above the diagonal.

1	1	0	0	0
5	2	8	0	0
0	8	3	2	0
0	0	4	1	5
0	0	0	7	9

* {1, 2, 3, 1, 9} = diagonal elements.
* {5, 8, 4, 7, 1, 8, 2, 5} = immediate diagonal elements.

Row wise mapping;
{1, 1, 5, 2, 8, 8, 3, 2, 4, 1, 5, 7, 9}

Column wise mapping;
{1, 5, 1, 2, 8, 8, 3, 4, 2, 1, 7, 5, 9}

Diagonal wise mapping;
{1, 8, 2, 5, 1, 2, 3, 1, 9, 5, 8, 4, 7}

* Representation of Sparse Matrix :-

A Sparse Matrix can be represented by using 2 representations :-

i) Array Representation
ii) Linked Representation

(1) **Array Representation** = In this representation, we consider only non-zero values along with their row & column index values.
In this representation, the 0^{th} row stores the total no. of columns & total no. of rows & the total no. of non-zero values in the sparse matrix.
For eg :-

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

→ 5 x 6 size matrix, containing 6 number of non zero values.

This matrix can be represented as :-

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2

row no. 0 ke column no. 4 ← me value hai 9.

row no. 3 ke column no. ← 5 me value hai 5.

0^{th} row store → the total no. of rows, total no. of columns & total no. of non-zero values in our 5x6 matrix.

(2) Linked Representation = In Linked representation, we use a linked list data structure to represent sparse matrix. In this linked list we use 2 nodes, i.e.; Header Node & Element Node. Header Node consist of 3 fields & element node consist of 5 fields.

Header Node

Index	Value
Down	Right

Element Node

Row	Column	Value
Down/up	Right	

* Array Representation Code of Sparse Matrix:

```
#include <stdio.h>
int main()
{
    int sparse_matrix[5][4] =
      {
           {0, 0, 3, 0},
           {0, 0, 5, 7},
           {0, 0, 0, 0},
           {0, 2, 6, 0},
           {4, 0, 0, 0}
      };
    int size = 0;
```

```
for(int i = 0; i < 5; i++)
{
  for(int j = 0; j < 4; j++)
  {
    if(sparse_matrix[i][j] != 0)
    {
        size++;
    }
  }
}

int new_matrix[3][size];
int R = 0;
for(int i = 0; i < 5; i++)
{
  for(int j = 0; j < 4; j++)
  {
    if(sparse_matrix[i][j] != 0)
    {
        new_matrix[0][R] = i;
        new_matrix[1][R] = j;
        new_matrix[2][R] = sparse_matrix[i][j];
        R++;
    }
  }
}
for(int i = 0; i < 3; i++){
    for(int j = 0; j < size; j++){
        printf("%d ", new_matrix[i][j]);
    }
    printf("\n");
}
return 0;
}
```