SPARSE MATRIX.

→ Sparse Matrix → is a matrix that has large number of elements with zero value. In order to efficiently utilize the memory, specialized algorithms and data structure that take advantage of the sparse structure should be used.

→ If we apply the operations using standard matrix structures and algorithms to sparse matrices, then the execution will slow down and matrix will consume large amount of memory.

– Sparse data can be easily compressed → which reduce memory usage.

There are two types of Sparse Matrices

① In the first type of sparse matrix, all elements above the main diagonal have a zero value. This type of Sparse Matrix is also called a (lower) triagonal matrix

↳ all non-zero value appear below the diagonal.

here $A_{ij} = 0$ where $i < j$.

↳ An $n \times n$ lower-triangular matrix A has one non-zero element in the first row, two non-zero elements in the second row likewise n – non-zero element in the n^{th} row.

→ To store a lower-triangular matrix efficiently in the memory, we can use a one-dimensional array which stores only non-zero elements

The mapping b/w a two-dimensional matrix and a one-dimensional array can be done in any one of the following ways →

(a) Row-wise Mapping → here the contents of array A[ ] will be

{1, 5, 3, 2, 7, -1, 3, 1, 4, 2, -9, 2, -8, 1, 7}

(b) Column-wise Mapping → here the contents of array A[ ] will be

{1, 5, 2, 3, -9, 3, 7, 1, 2, -1, 4, -8, 2, 1, 7}

$$\begin{bmatrix} 1 & & & & \\ 5 & 3 & & & \\ 2 & 7 & -1 & & \\ 3 & 1 & 4 & 2 & \\ -9 & 2 & -8 & 1 & 7 \end{bmatrix}$$

Lower Triangular Matrix

(b) There is a upper-triangular matrix

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ & 3 & 6 & 7 & 8 \\ & & -1 & 9 & 1 \\ & & & 9 & 2 \\ & & & & 7 \end{bmatrix}$$

⇒ In a upper triangular matrix $A_{i,j} = 0$ where $i > j$. An $n \times n$ upper triangular matrix A has n non-zero elements in the first row, ~~n+~~ ~~n+~~ $n-1$ non-zero element in second row and likewise one non-zero element in the nth row.

ⓒ There is a another variant of a sparse matrix, in which elements with a non-zero value can appear only on the diagonal or immediately above or below the diagonal.

This type of Matrix is called "TRI-DIAGONAL MATRIX".

Hence in a tridiagonal matrix $A_{ij} = 0$ where $|i-j| > 1$.

In a tridiagonal matrix, if elements are present on

(a). The main diagonal, it contains non-zero elements for $i=j$. In all, there will be n elements.

(b). Below the main diagonal, it contains non-zero elements for $i=j+1$. In all, there will be $n-1$ elements.

(c). Above the main diagonal, it contains non-zero elements for $i=j-1$. In all, there will be $n-1$ elements.

MULTIDIMENSIONAL-ARRAYS

A two dimensional array are collection of homogeneous elements, where the elements are ordered in number of rows and columns. An example of an $m \times n$ matrix, where m denotes the no. of rows and n denotes the number of columns

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & & a_{mn} \end{bmatrix}_{m \times 4}$$

Memory Representation of a Matrix

→ Like One-dimensional arrays, matrices are also stored in Contagious Memory locations.

There are two conventions of storing any matrix in the memory :→

(1). $\boxed{\text{Row-major Order}}$ (2). $\boxed{\text{Column-major order.}}$

(1) In this, elements of a matrix are stored, row by row basis, that is, all the elements in the first row, then in the second row, & so on.

(2) In this, elements are stored column by column, that is, all the elements in the first column stored in their order of rows, then in the second column, third column, and so on.

Reference of elements in a matrix

Logically, a matrix appears as 2-D → but physically it is stored in a linear fashion ⟶ So, in order to map from logical view to physical structure, we need an indexing formula.

The indexing formula, for different orders are stated below:→

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}_{3\times 4}$$

Row-major Order

Assume that the base address is the first location of the memory that is, 1. So, the address of a_{ij} will be obtained as

Address (a_{ij}) = Storing all the elements in first $(i-1)$th rows
+
the no. of elements in ith row up to the jth column

Address (a_{ij}) = $\boxed{= (i-1) \times n + j}$ $(3-1)\times 4 + 2$

So, for the matrix $A_{3\times4}$, the location of a_{32} will be calculated as 10. Instead of considering the base address to be 1, if it is at M, then above formula modified as

* $\boxed{\text{Address } (a_{ij}) = M + (i-1)\times n + j - 1}$

Column-Major Order

Address (a_{ij}) = Storing all the elements in the first $(j-1)$th columns + the number of elements in the jth column up to the ith row

3×4

a_{32} a_{14}

$= (j-1) \times m + i$ $2\times$ ~~$(3-1)\times 4 + 2$~~ $(4-1)\times 3 + 1$

$3\times4+2$ $9+1$

Considering the Base address at M instead of 1, the above formula will get modified

$\boxed{\text{Address } (a_{ij}) = M + (j-1) \times m + i - 1}$

Sparse Matrix.

A sparse matrix is a Two-Dimensional Array where the majority of the elements have the value null.

Sparse Matrices (Symmetric)

- Triangular Matrices
  - Lower
    - Lower Left
    - Lower Right
  - Upper
    - Upper Left
    - Upper Right
- Band Matrices
  - Diagonal
  - Tridiagonal
  - $\alpha\beta$-band.

→ In a large no. of application, sparse matrices are involved. So far as the storage of a sparse matrix is concerned, storing of null elements is nothing but wastage of memory. So, we should devise a Technique such that only non-null element will be stored.

Memory representation of a lower-triangular matrix.

$$\begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ a_{31} & a_{32} & a_{33} & \\ \vdots & \vdots & \vdots & \\ a_{n1} & a_{n2} & a_{n3} \cdots & a_{nn} \end{bmatrix}_{n \times n}$$

Row-major Order

Acc. to row-major order, the address of any element, ~~the address~~ a_{ij}, $1 \le i, j \le n$, can be obtained as

Address (a_{ij}) = No. of elements upto the a_{ij} element

= Total no. of elements in the first $i-1$ rows

+

No. of elements upto the j^{th} column in the i^{th} row

$$= 1 + 2 + 3 + \cdots + (i-1) + j$$

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

if the starting location of the first element, that is, of a_{11} is M, then the address of a_{ij}, $1 \le i, j \le n$ can be obtained as

$$\text{Address}(a_{ij}) = M + \frac{i(i-1)}{2} + j - 1$$

Column Major Order :→

According to column major order, the address of any element a_{ij}, $1 \le i, j \le n$ can be obtained as

Address (a_{ij}) = Number of elements up to the a_{ij} element

= Total no. of elements in the first $j-1$ columns

+ number of elements upto the ith row in the jth column

$$= [n + (n-1) + (n-2) + \cdots + (n-j+2)] + (i-j+1)$$

$$= \{n \times (j-1) - [1+2+3+\cdots+(j-2)+(j-1)] + i\}$$

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

$$= (j-1) \times (n - j/2) + i$$

If the starting location of the first element (that is, of a_{11}) is M, then the address of a_{ij}, $1 \le i, j \le n$ will be

$$\boxed{\text{Address}(a_{ij}) = M + (j-1) \times (n - j/2) + i - 1}$$

Memory Representation of an upper-triangular matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ & a_{22} & a_{23} & \cdots & a_{2n} \\ & & a_{33} & \cdots & a_{3n} \\ & & & & \vdots \\ & & & & a_{nn} \end{bmatrix}_{n\times n}$$

Row major Order.

Acc. to row-major order, the address of any element a_{ij}, $1 \le i, j \le n$ can be obtained

Address (a_{ij}) = No. of elements up to the a_{ij} element
= Total no. of elements in the first $(i-1)$ rows
+
No. of elements up to the j^{th} column in the i^{th} row

$$= n + (n-1) + (n-2) + \cdots + (n-i+2) + (j-i+1)$$

$$= n \times (i-1) - [1+2+3+\cdots+(i-2)+(i-1)] + j$$

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

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

If the starting location of the first element, i.e of a_{11} is M, then the address of a_{ij}, $1 \le i, j \le n$ will be

$$\boxed{\text{Address } (a_{ij}) = M + (i-1) \times \left(n - \frac{i}{2}\right) + j - 1}$$

Column-major Order:-

Address (a_{ij}) = No. of element up to the a_{ij} elements
= Total no. of element in the first $(j-1)$ column
+ no. of element upto the i^{th} row in the j^{th} column

$$= [1+2+3+\cdots+(j-1)] + i$$

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

If the starting location of the first element,

$$\boxed{\text{Address } (a_{ij}) = M + \frac{j(j-1)}{2} + i - 1}$$

Memory Representation of Diagonal Matrix

In sparse matrix having the elements only on the diagonal, the following points are evident :→

No. of elements in an $n \times n$ square diagonal matrix $= n$

Any element a_{ij} can be stored in memory using the formula

Address $(a_{ij}) = i$ [or j]

Memory Representation of a Tridiagonal Matrix

$$\begin{bmatrix} a_{11} & a_{12} & & & & & \\ a_{21} & a_{22} & a_{23} & & & & \\ & a_{32} & a_{33} & a_{34} & & & \\ & & a_{43} & a_{44} & a_{45} & & \\ & & & \vdots & \vdots & & \\ & & & & a_{(n-1)(n-2)} & a_{(n-1)(n-1)} & a_{(n-1)(n)} \\ & & & & & a_{n(n-1)} & a_{nn} \end{bmatrix}$$

Row-major order :→

Acc. to row major order, the address of any element a_{ij}, $1 \le i, j \le n$ can be obtained as

Address (a_{ij}) = No. of elements upto a_{ij} elements

= Total no. of elements in the first $(i-1)$ rows + no. of elements upto the j^{th} column in the i^{th} row

$= \{2 + [3 + 3 + \cdots + \text{upto } (i-2) \text{ terms}]\} + (j - i + 2)$

$= 2 + (i-2) \times 3 + j - (i-2)$

$= 2 + 2 \times (i-2) + j$

If the starting location of first element, i.e a_{11} is M, then address of a_{ij} $1 \le i, j \le n$ will be

$$\boxed{\text{Address}(a_{ij}) = M + 2 \times (i-2) + j + 1}$$

Column major Order

Acc. to column major-order, the address of any element a_{ij}, $1 \le i, j \le n$ can be obtained as,

Address (a_{ij}) = No. of elements upto a_{ij} element
= total no. of element in the first $(j-1)$ column
+ no. of element upto ith row in the jth column.

$$= \{2 + [3 + 3 + \cdots + \text{upto } (j-2) \text{ terms}]\} + (i - j + 2)$$

$$= 2 + (j-2) \times 3 + i - (j-2)$$

$$= 2 + 2 \times (j-2) + i$$

If the starting location of the first element, i.e of a_{11} is M, then the address of a_{ij}, $1 \le i, j \le n$ will be

$$\text{Address}(a_{ij}) = M + 2 \times (j-2) + i + 1$$

NOTE → The formula for row/column-major order is symmetric and one can be obtained from the others by interchanging i and j.