ATME College of Engineering

Department of Computer Science and Engineering

Academic Year: 2024-25

Group Assignment	
Subject: Data Structure and Applications	
Subject Code: BCS304	**SEM & Section:** 3rd B

GROUP ACTIVITY

Group No: 5

Submitted BY:

PUNEETH J	USN 4AD23CS083
R RISHUPRASAD	USN 4AD23CS084
RAGHUNANDAN A S	USN 4AD23CS085
RAKSHITHA R	USN 4AD23CS086
RAKSHITHA S	USN 4AD23CS087

TITLE: SPARE MATRIX USING ARRAYS

I. INTRODUCTION

Sparse matrices are matrices that have most of their elements as zero. Efficient storage and processing of such matrices is achieved by only storing non-zero elements along with their row and column indices. Here's a detailed explanation of sparse matrices using arrays:

1. Detailed Explanation

A sparse matrix is a matrix that has a large number of zero elements compared to non-zero elements. Representing sparse matrices in a conventional 2D array is memory-inefficient. Instead, they are represented using specialized storage techniques like arrays, linked lists, or other structures.

Benefits of Sparse Matrix Representation:

Memory Efficiency: Only non-zero elements are stored, reducing memory usage.

Faster Processing: Operations like addition and multiplication are quicker since most elements are zero

Common Representations:

1. Triplet Representation (3-array representation):

Three arrays are used:

Row indices of non-zero elements.

Column indices of non-zero elements.

Values of non-zero elements.

2. Compressed Sparse Row (CSR) and Compressed Sparse Column (CSC):

Used for computational efficiency in large-scale applications.

II. GENERAL SYNTAX

Triplet Representation:

Let a be the 2D sparse matrix of size m x n.

Use three arrays:

row[]: Stores row indices of non-zero elements.

col[]: Stores column indices of non-zero elements.

val[]: Stores values of non-zero elements.

Pseudo Representation:

```
row[i] = row index of i-th non-zero element
col[i] = column index of i-th non-zero element
val[i] = value of i-th non-zero element
```

III. EXAMPLE

Consider the matrix:

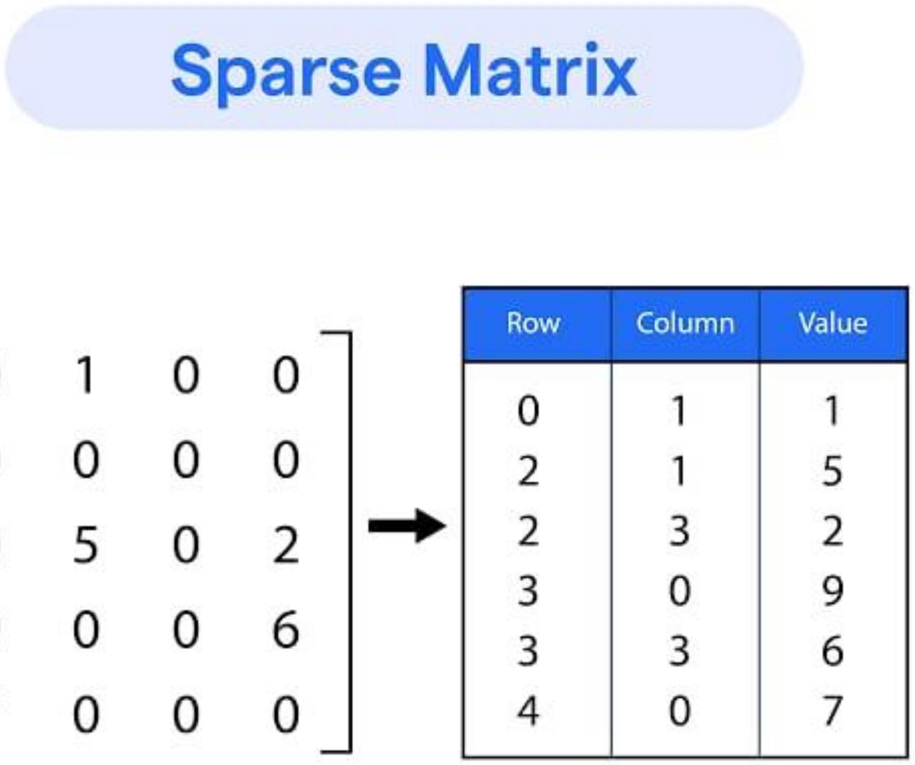
```
 0  0  3
 0  5  0
 8  0  0
```

Triplet representation:

row = {0, 1, 2}

col = {2, 1, 0}

val = {3, 5, 8}



Sparse Matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 2 \\ 9 & 0 & 0 & 6 \\ 7 & 0 & 0 & 0 \end{bmatrix} \rightarrow$$

Row	Column	Value
0	1	1
2	1	5
2	3	2
3	0	9
3	3	6
4	0	7

IV. C PROGRAM with OUTPUT

```
#include <stdio.h>

void displayTriplet(int row[], int col[], int val[], int size) {
   printf("Row  Col  Value\n");
   for (int i = 0; i < size; i++) {
      printf("%3d  %3d  %3d\n", row[i], col[i], val[i]);
   }
}

int main() {
   int matrix[3][3] = {
      {0, 0, 3},
      {0, 5, 0},
      {8, 0, 0}
   };
   int row[10], col[10], val[10];
   int k = 0;

   printf("Original Matrix:\n");
   for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
         printf("%3d ", matrix[i][j]);
         if (matrix[i][j] != 0) {
            row[k] = i;
            col[k] = j;
            val[k] = matrix[i][j];
            k++;
         }
      }
      printf("\n");
   }

   printf("\nTriplet Representation:\n");
   displayTriplet(row, col, val, k);

   return 0;
}
```

Possible Output:

```
Original Matrix:
  0   0   3
  0   5   0
  8   0   0

Triplet Representation:
Row  Col  Value
  0    2    3
  1    1    5
  2    0    8
```

V. APPLICATIONS

1. Scientific Computing: Representing systems of linear equations, graphs, and adjacency matrices efficiently.

2. Image Processing: Sparse representations are used in areas where most pixels are black or white.

3. Machine Learning: Storing data with a large number of features (e.g., text datasets in NLP).

4. Network Analysis: Representing sparse networks like social networks or communication graphs.

5. Database Storage: Optimizing storage of datasets with many missing values.