

DATA STRUCTURES

[HOME](#)[SUBJECTS](#)[DOWNLOADS](#)[AUTHORS](#)[CONTACT US](#)

UNIT 1

[Introduction to](#)[Algorithm](#)[Performance](#)[Analysis](#)[Space](#)[Complexity](#)[Time](#)[Complexity](#)[Asymptotic](#)[Notations](#)[Linear & Non-](#)[Linear Data](#)[Structures](#)[Single Linked](#)[List](#)[Circular](#)[Linked List](#)[Double Linked](#)[List](#)[Arrays](#)[Sparse Matrix](#)

UNIT 2

[Stack ADT](#)[Stack Using](#)[Array](#)[Stack Using](#)[Linked List](#)[Expressions](#)[Infix to Postfix](#)[Postfix](#)[Evaluation](#)[Queue ADT](#)[Queue Using](#)[Array](#)

Sparse Matrix



What is Sparse Matrix?

In computer programming, a matrix can be defined with a 2-dimensional array. Any array with 'm' columns and 'n' rows represents a mXn matrix. There may be a situation in which a matrix contains more number of ZERO values than NON-ZERO values. Such matrix is known as sparse matrix.

Sparse matrix is a matrix which contains very few non-zero elements.

When a sparse matrix is represented with 2-dimensional array, we waste lot of space to represent that matrix. For example, consider a matrix of size 100 X 100 containing only 10 non-zero elements. In this matrix, only 10 spaces are filled with non-zero values and remaining spaces of matrix are filled with zero. That means, totally we allocate $100 \times 100 \times 2 = 20000$ bytes of space to store this integer matrix. And to access these 10 non-zero elements we have to make scanning for 10000 times.

Sparse Matrix Representations

A sparse matrix can be represented by using TWO representations, those are as follows...

1. Triplet Representation
2. Linked Representation

Triplet Representation

In this representation, we consider only non-zero values along with their row and column index values. In this representation, the 0th row stores total rows, total columns and total non-zero values in the matrix.

For example, consider a matrix of size 5 X 6 containing 6 number of non-zero values. This matrix can be represented as shown in the image...

Queue Using
 Linked List
 Circular Queue
 Double Ended
 Queue
UNIT 3
 Tree -
 Terminology
 Tree
 Representations
 Binary Tree
 Binary Tree
 Representations
 Binary Tree
 Traversals
 Threaded
 Binary trees
 Max Priority
 Queue
 Max Heap
 Introduction to
 Graphs
 Graph
 Representations
 Graph
 Traversal - DFS
 Graph
 Traversal - BFS
UNIT 4
 Linear Search
 Binary Search
 Hashing
 Insertion Sort
 Selection Sort
 Radix Sort
 Quick Sort
 Heap Sort
 Comparison of
 Sorting
 Methods
UNIT 5
 Binary Search
 Tree
 AVL Trees

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



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

In above example matrix, there are only 6 non-zero elements (those are 9, 8, 4, 2, 5 & 2) and matrix size is 5 X 6. We represent this matrix as shown in the above image. Here the first row in the right side table is filled with values 5, 6 & 6 which indicates that it is a sparse matrix with 5 rows, 6 columns & 6 non-zero values. Second row is filled with 0, 4, & 9 which indicates the value in the matrix at 0th row, 4th column is 9. In the same way the remaining non-zero values also follows the similar pattern.

Linked Representation

In linked representation, we use linked list data structure to represent a sparse matrix. In this linked list, we use two different nodes namely **header node** and **element node**. Header node consists of three fields and element node consists of five fields as shown in the image...

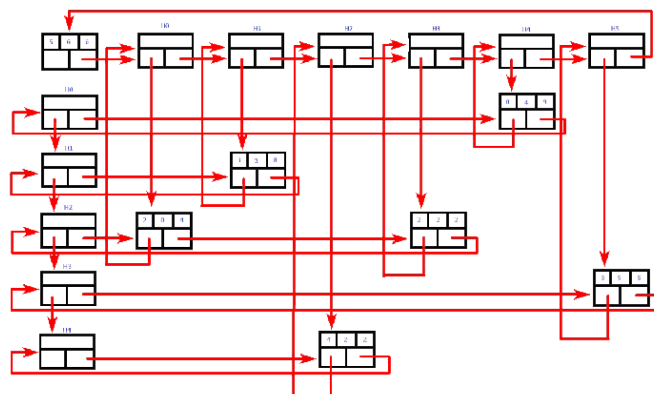
Header Node



Element Node



Consider the above same sparse matrix used in the Triplet representation. This sparse matrix can be represented using linked representation as shown in the below image...



In above representation, H0, H1,...,H5 indicates the header nodes which are used to represent indexes. Remaining nodes are used to represent non-zero elements in the matrix, except the very first node which is used to represent abstract information of the sparse matrix (i.e., It is a matrix of 5 X 6 with 6 non-zero elements).

B - Trees
Red - Black
Trees
Splay Trees
Comparison of
Search Trees
Knuth-Morris-
Pratt
Algorithm
Tries

In this representation, in each row and column, the last node right field points to it's respective header node.



[About Us](#) | [Contact Us](#)

Website designed by RAJINIKANTH B