**MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL**

Submitted By: Yashwant Patidar

Scholar Number: 191112243

Section: CSE 2

Subject: Data Structures - CSE 212

**1. Representation Of Sparse Matrix:**

A matrix is a two-dimensional data object made of m rows and n columns, therefore having total m x n values. If most of the elements of the matrix have **0 values**, then it is called a sparse matrix.

The two representations are:-

1. Array Representation

2. Linked Representation

ARRAY REPRESENTATION :-

2D array is used to represent a sparse matrix in which there are three rows named as -

* • **Row:** Index of row, where non-zero element is located
* • **Column:** Index of column, where non-zero element is located
* • **Value:** Value of the non zero element located at index – (row, column)

LINKED REPRESENTATION :-

In linked representation, each node has four fields. These four fields are defined as:

* • **Row:** Index of row, where non-zero element is located
* • **Column:** Index of column, where non-zero element is located
* • **Value:** Value of the non zero element located at index – (row, column)
* • **Next right node:** Address of the next right node in the next column
* • **Next down node:** Address of the next below node in the next row

**Comparing both Array and Linked representation in doing the following operations:**

1. Searching for an element

Time Complexity:

In the array representation of the sparse matrix, since the elements would be unsorted, we would have to linearly scan through the whole matrix to find the element. For searching an element in array representation, it has a linear time complexity. So, O(n) complexity.

In linked list representation, since the traversal would be linear, the time complexity would be linear as well. So, O(n) complexity.

Space Complexity:

While searching, only a fixed number of variables are to be used, and none of it is dependent on the number of elements in sparse matrix, it can be said that the space complexity of array representation of sparse matrix is constant. S(p)>=1;

In the linked list representation as well, since the searching parameters don’t depend on the number of elements in sparse matrix, it has a constant space complexity. S(p) >=1;

2. Inserting a new element:

Time Complexity:

In array representation, we will have to move all the elements by 1 position forward where the element is to be stored. In the worst case, we may have to move all the n elements. So, for inserting a new element, it has a linear time Complexity. O(n).

In linked list representation, after we have found the node after which we have to insert the element, we can perform the insertion in constant time. So, insertion in linked list representation of sparse matrix is a constant time operation. O(1).

Space Complexity:

Since there is no new memory being added to the array after the creation, it has a constant space complexity. It has S(p)>=1.

We need to add only one node to the existing linked list representation, so it has a constant space complexity. It has S(p)>=1.

3. Deleting a new elements

Time Complexity

In array representation, we will have to move all the elements by 1 position backward where the element is to be deleted. In the worst case, we may have to move all the n elements. So, for deleting an element, it has a linear time Complexity. O(n).

In linked list representation, after we have found the node which has to be deleted, we can perform the deletion in constant time. So, deletion in linked list representation of sparse matrix is a constant time operation. O(1).

Space Complexity

For the array representation, the deletion operation, there is no need for a new memory space, it has a constant space complexity. It has S(p)>=1.

We remove a node from the existing linked list representation, so it has a constant space complexity. It has S(p)>=1.

4. Adding two matrices of same dimension

Time Complexity

For adding two sparse matrices, we will create a new empty sparse matrix. Then, we will traverse through both the matrices, and if there is already an element there, we add the value to the previous element. The time complexity would be O(t1+t2) where t1 and t2 are the number of non zero elements in the two matrices.

In linked lists, we traverse linearly through both the matrices, and add a new element to the third list if no element is found. If the element exists, we add the value to the existing value. So, it has a linear time complexity.

Space Complexity

In the array representation, since we would have to create a new linked list to store the sum, the S(p)>=max(t1, t2), where t1 and t2 is the number of non zero elements in the matrices.

For linked list, the complexity can be written as S(p)>=(max(t1,t2) + m +n). This is a linear space complexity.