Sparse Matrix

Def:→ Matrices with a relatively high proportion of zero entries are called sparse matrix.

Representation of Sparse matrix:→

Tuple Method:→ A sparse matrix can be conveniently stored in the memory using 3 tuple method. Using this method, only the non-zero entries from the given matrix are stored in 3 tuples. The 3-tuples are - row, column & value.

Consider the following sparse matrix with 3 row & 4 columns.

$$\begin{pmatrix} 15 & 0 & 0 & 21 \\ 0 & 0 & 25 & 0 \\ 19 & 0 & 0 & 29 \end{pmatrix}$$

The 3 tuple representation of the above matrix will be:

	0	1	2
0	(No. of row) 3	(no. of col) 4	(value) 5
1	0	0	15
2	0	3	21
3	1	2	25
4	2	0	19
5	2	3	29

② **Transpose of a Sparse Matrix:→**

Procedure – Transpose (A, B)

[A is a sparse matrix in tuple form & B is set to be its transpose] [m → no. of row, n – no of col, t → no of non-zero value]

S1:→ $(m, n, t) \leftarrow (A(0,0), A(0,1), A(0,2))$

S2:→ $(B(0,0), B(0,1), B(0,2)) \leftarrow (n, m, t)$

S3:→ If $t \leq 0$ then return

S4:→ $q \leftarrow 1$

S5:→ for col ← 0 to n−1 do

S6:→ for p ← 1 to t do

S7:→ If $A(p,1)$ = col then

S8:→ $[B(q,0), B(q,1), B(q,2)] \leftarrow [A(p,1), A(p,0), A(p,2)]$

S9:→ $q \leftarrow q+1$

end

end

S10:→ End Transpose.

Ex:→

$$\begin{bmatrix} 15 & 0 & 0 & 22 & 0 & -15 \\ 0 & 11 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 \end{bmatrix}$$

A

	0	1	2
0	6	6	8
1	0	0	15
2	0	3	22
3	0	5	-15
4	1	1	11
5	1	2	3
6	2	3	1
7	4	0	10
8	5	2	8

B

	0	1	2
0	6	6	8
1	0	0	15
2	0	4	10
3	1	1	11
4	2	1	3
5	2	5	8
6	3	0	22
7	3	2	1
8	5	0	-15

Time complexity of the above algorithm is $O(nt)$.

4

Fast_Transpose (A, B)

[A is a sparse matrix in tuple form & B is set to be its transpose]

declare S[0 : n-1] , T[0 : n-1]

S1 :-> (m, n, t) ← (A(0,0), A(0,1), A(0,2))

S2 :-> [B(0,0), B(0,1), B(0,2)] ← (n, m, t)

S3 :-> If t ≤ 0 then return.

S4 :-> For i := 0 to n-1 do S[i] ← 0 end

S5 :-> for i = 1 to t do

S6 :-> S[A(i,1)] ← S[A(i,1)] + 1

end

S7 :-> T[0] ← 1

S8 :-> For i ← 1 to n-1 do

S9 :-> T[i] ← T[i-1] + S[i-1]

end

S10 :-> for i ← 1 to t do

S11 :-> j ← A(i, 1)

S12 :-> [B(T(j), 0), B(T(j), 1), B(T(j), 2)] ← [A(i,1), A(i,0), A(i,2)]

S13 :-> T(j) ← T(j) + 1

end

S14 :-> End. Transpose.