ISSN:0975-9646
Sanil Shanker KP et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (5) , 2016, 2347-2348

Time Complexity of Matrix Transpose Algorithm using Identity Matrix as Reference Matrix

Sanil Shanker KP[1] , Mohammed Shameer MC[2]

[1,2]Dept. of Computer Science
Farook College, Kozhikode, India.

Abstract- This paper presents the time complexity of matrix transpose algorithm using identity matrix as reference matrix. We computed the time complexity of the algorithm as O(*mn*).

Keywords: Identity matrix, Reference matrix, Sanil's Matrix Transpose.

I. INTRODUCTION

Transpose of the matrix can be obtained by combining the characteristics of logical AND (∧) with logical OR (∨) operations [1, 2]. In Sanil's matrix transpose algorithm, the identity matrix acts as the kernel of the transformation [3]. For example, let the matrix $A_{(3 \times 4)}$ be

17	2	13	7
41	11	29	19
19	3	23	11

The transformation can computed as:

Input: $\mathbf{A_{(3 \times 4)}}$ logical **AND** $\mathbf{I_3}$

17	2	13	7	^	**1**	**0**	**0**
41	11	29	19	^	**0**	**1**	**0**
19	3	23	11	^	**0**	**0**	**1**

↓ ↓ ↓

17	41	19
2	11	3
13	29	23
7	19	11

Output: $\mathbf{A^T_{(4 \times 3)}}$

Here, identity matrix acts as the kernel to find the transpose.

II. TIME COMPLEXITY

Let $A_{(m \times n)}$ and $B_{(m \times m)}$ be the input matrix of order (m x n) and the reference matrix of order (m x m) respectively. The value of c_{11} can be computed from the Figure- 1, as $c_{11} := (a_{11} * b_{11}) + (a_{21} * b_{21}) + (a_{31} * b_{31})$.

$$
\begin{array}{ccccc|ccc}
\multicolumn{5}{c}{\mathbf{A}_{(m \times n)}} & \multicolumn{3}{c}{\mathbf{B}_{(m \times m)}} \\
a_{11} & a_{12} & a_{13} & a_{14} & . & b_{11} & b_{12} & b_{13} \\
a_{21} & a_{22} & a_{23} & a_{24} & . & b_{21} & b_{22} & b_{23} \\
a_{31} & a_{32} & a_{33} & a_{34} & . & b_{31} & b_{32} & b_{33} \\
 & & & & & \downarrow & \downarrow & \downarrow \\
 & & & & & c_{11} & c_{12} & c_{13} \\
 & & & & & c_{21} & c_{22} & c_{23} \\
 & & & & & c_{31} & c_{32} & c_{33} \\
 & & & & & c_{41} & c_{42} & c_{43}
\end{array}
$$

Fig. 1 $\mathbf{A}_{(n \times m)}{}^{\mathbf{T}} := \mathbf{C}_{(n \times m)}$

To compute one cell value, there exists 'm' multiplications and 'm-1' additions. For the transformation, $c_{nm} \leftarrow a_{mn}$, the computational time is O(m). If there exists 'm' rows, time will be O(m) + O(m) +……..m times = O(m^2). For 'n' columns, the computational time is O(nm^2).

In the case of identity matrix as reference matrix, ($a_{i = m, j = n} * I_{i = m, j = m}$) exists and other will be zero (Figure- 2) [2]. This implies the time for one multiplication operation will be O(1). If there exists 'm' rows, time will be O(1) + ……..m times = O(m). In general, for 'n' columns, time = O(mn).

$$\mathbf{A}_{(m \times n)} \qquad \qquad \mathbf{I}_{(i = m,\, j = m)}$$

$$\begin{array}{ccccc|ccc}
a_{11} & a_{12} & a_{13} & a_{14} & . & I_{11} & 0 & 0 \\
a_{21} & a_{22} & a_{23} & a_{24} & . & 0 & I_{22} & 0 \\
a_{31} & a_{32} & a_{33} & a_{34} & . & 0 & 0 & I_{33} \\
 & & & & & \downarrow & \downarrow & \downarrow \\
 & & & & & c_{11} & c_{12} & c_{13} \\
 & & & & & c_{21} & c_{22} & c_{23} \\
 & & & & & c_{31} & c_{32} & c_{33} \\
 & & & & & c_{41} & c_{42} & c_{43}
\end{array}$$

Fig. 2 $\mathbf{A}_{(n \times m)}{}^{\mathbf{T}} := \mathbf{C}_{(n \times m)}$

III. Summary

The computational time of matrix transpose algorithm using identity matrix as reference matrix is $O(mn)$. Suppose, if the given matrix is a square matrix, the running time will be $O(n^2)$.

References

[1] Sanil Shanker KP, *An Algorithm to Transpose Zero- One Matrix.* Int. Journal of Com. Sci. and Inf. Tech, Vol. 7 (4), 2016 , 1960- 1961.

[2] Sanil Shanker KP, *Will the reference Matrix act as the Kernel of Matrix Transformation?,* Int. Journal of Math. Tre. and Tech, Vol. 36 (1), 2016, 80- 81.

[3] Mohammed Shameer. MC, Sanil Shanker. K. P, *Java Implementation of Sanil's Matrix Transpose,* Int. Journal of Com. Sci. and Inf. Tech. Vol. 7(5), 2016, 2145- 2146.