Java Implementation of Sanil’s Matrix Transpose Algorithm
Mohammed Shameer MC¹, Sanil Shanker KP²

¹²Dept. of Computer Science
Farook College, Kozhikode, India.

Abstract- In 2016, Sanil put forward a method for matrix transformation [1]. This paper illustrates how matrix transpose can be done by using identity matrix as reference matrix. For simulating the algorithm, the program has been written in Java under Linux platform.

Keywords: Matrix Transformation, Matrix Transpose, Identity Matrix.

I. INTRODUCTION
The transpose of a matrix is obtained by interchanging the rows and columns. Here, we compute matrix transpose by combining the characteristics of logical AND with logical OR operation [1, 2]. For example, consider the matrix A of order m x n, where m = 2 and n = 3.

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix}
\]

Let the reference matrix be the identity matrix I₂

\[
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

The input matrix A operates logical AND with reference matrix I₂ gives Aᵀ with the cell values Wᵢᵣ.

\[
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix}
\land
\begin{pmatrix}
1 & 0 \\
0 & 1
\end{pmatrix}
\]

Identity Matrix (I₂)

The value of Wᵢᵣ can be computed as,

\[\sum_{i=1}^{m} Wᵢᵣ \quad i=1, m \land \quad j=1, n\]

This gives the transpose of the matrix A of order n x m as the output. That is, Aᵀ₂₃ ← A₂₃, I₂

\[
\begin{pmatrix}
1 & 4 \\
2 & 5 \\
3 & 6
\end{pmatrix}
\]

II. JAVA IMPLEMENTATION
Java implementation of the algorithm can be subdivided into three steps. The program takes a matrix of order (m x n) and produces another matrix of order (n x m).

1. Reading the order of the matrix and its elements. The given matrix is read to a two dimensional array with m rows and n columns. For reading data from the console (keyboard), the scanner class is used. Scanner is a simple text scanner that automatically parses the read data to primitive data types like integer, floating types etc using regular expressions. This class is included in the java util package and hence it should be imported to the program.

2. Generating the reference matrix A two dimensional array is allocated for the reference matrix. This identity matrix has m rows and columns; where m is the number of rows in the given matrix.

3. Algorithm implementation The transpose matrix is generated with the help of three nested loops. The outer loop iteration corresponds to the row and middle loop to columns. Elements in the corresponding location are multiplied and the terms are added in the inner loop.

The source code of the program is attached.

```java
import java.util.*;

class transform {
    public static void main(String args[]) {
        int i,j,k,m,n;  // variable declaration
        System.out.println("Enter the order");
        Scanner s = new Scanner(System.in);
        m= s.nextInt();
        n=s.nextInt();

        int[][] a = new int[m][n];

        //reading the order of the matrix to m,n
        m= s.nextInt();
        n=s.nextInt();

        //dynamic memory allocation to matrix A
        int[][] a = new int[m][n];
```
System.out.println("Enter the matrix");
// reading matrix elements to the 2D array
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        a[i][j]=s.nextInt();
    }
}
// creating identity reference matrix r of
// order m after dynamic memory allocation.
int[][] r = new int[m][m];
for(i=0;i<m;i++){
    for(j=0;j<m;j++){
        if(i==j)
            r[i][j]=1;
        else
            r[i][j]=0;
    }
}
// displaying the given matrix A
System.out.println("Given matrix");
for(i=0;i<m;i++){
    for(j=0;j<n;j++){
        System.out.print(a[i][j] + "   ");
    }
    System.out.println();
}
// algorithm implementation
System.out.println("Transpose Matrix");
int s1=0;
for(j=0;j<n;j++){
    for(i=0;i<m;i++){
        s1=0; // resetting the sum to zero
        for(k=0;k<m;k++){
            //multiplying corresponding elements in
            //the given matrix and reference matrix
            //and adding each term.
            s1 += a[i][j] * r[i][k];
        }
        System.out.print(s1+ " ");
    }
    System.out.println();
}
The program is tested matrices of various orders under
jdk versions 1.6 and 1.7 for Linux and Windows
platforms. Sample output is as given:
Enter the order
3 2
Enter the matrix
1 5 4 6 9 7
Given matrix
1 5
4 6
9 7
Transpose Matrix
1 4 9
5 6 7

III SUMMARY
Java implementation of Sanil’s Matrix Transpose
algorithm has been explained in the paper. This
simulation technique can be implemented to extend a
way of research in Graphics Transformation.

REFERENCES
Int.J.of Computer Science and Information Technologies, Vol. 7
[2] Sanil Shanker KP, Will the Reference Matrix acts as the Kernel
of Matrix Transformation?, Int. J. of Mathematics Trends and
Technology, Vol.36, 1, August 2016.