A Technique to Find the Trace of Square Matrix

Khairunnisa K
Postgraduate and Research Department of Computer Science
Farook College (Autonomous)
Calicut, India

Abstract — This paper explains a method to compute the trace of a square matrix. We illustrate the method with random data and simulate the algorithm in JAVA language.

Keywords — Square matrix, Trace of matrix.

I. Introduction

Trace of an n-by-n square matrix is defined as the sum of main diagonal elements of matrix [1]. In this paper, we put forward a method to compute the trace of a matrix by setting a counter. Here the counter value increment until it reaches $(n+1)^{th}$ element and it add to the trace. This process continues until the positional value of the element equal to n^2.

II. Example

CONSIDER A SQUARE MATRIX OF ORDER (3 x 3)

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

1. Initialize trace, Tr to zero. First we select 1 and add it to the Tr, Set the counter to zero (c=0).
2. Input 2, counter increments (c=1). Check whether counter is n+1, if not, move to next element.
3. Input 3, counter increments (c=2). Check whether counter is n+1, if not, move to next element.
4. Input 4, counter increments (c=3). Check whether counter is n+1, if not, move to next element.
5. Input 5, counter increments (c=4). Check whether counter is n+1. Here counter value is equal to n+1, so element is added with Tr.
6. Continuous this process until all elements are encountered.

III. Algorithm

```
Step 1 :   Start.
Step 2 :   Declare variable c, k and t;
           Initialize c = -1 and t=0.
Step 3 :   Read order of the square matrix, n.
Step 4 :   For i = 0 to n*n
               Read element, k.
               If (c == (n+1)) then
                         $t = t + k$
                       c = 0
               else if (c ==  -1) then
                       t = k
                       c = 0
               end if
               increment c
Step 5 :   Print t
Step 6 :   Stop
```

IV. Java Implementation

The implementation of this algorithm can be done as follows:

```
//Source code of the program :
//program name : Trace.java
//input : Matrix (n×n)
//output: Trace of matrix

import java.util.*;

class Trace
{
public static void main(String[] arg)
{
   int c=-1, k, t=0; // variable declaration and
                         Initialization
   Scanner sc = new Scanner (System.in);
   System.out.println ("Enter order:");

int n = sc.nextInt(); //Reading order of the
                               square matrix
   System.out.println("Enter elemnts: ");
   for(int i=0;i<n*n;i++)
   {
         k = sc.nextInt(); //Reading elements

          if(c==(n+1)) //Checking for diagonal
                            element using counter
          {
                  t+=k;
                  c=0;
          }
         else if(c==-1) //Checking for the first
                               element
         {
             t=k;
                 c=0;
         }
          c++;
     }
     System.out.println("Trace of matrix " + t);
                          //Printing the Trace
   }
}
```

The program is tested with matrices of various orders.

For example the trace of a matrix of order 3 will be as:

```
Enter order:
3
Enter elemnts:
1 2 3
4 5 6
7 8 9
Trace of matrix  15
```

V. CONCLUSION

Here, we put forward a technique to compute trace of a square matrix. We illustrate the method with numerical data. In this method, we eliminate the need for nested loops.

REFERENCE

[1] P.M. Cohn, "Algebra" , 1 , Wiley (1982) pp. 336