Design And Analysis Of Algorithms-Assignment 2
Group-2

ADITYA RAJ(IIT2017005)
MOHIT PAHUJA(IIT2017007)
MAYANK MRINAL(IIT2017006)
ARASHPREET SINGH (IIT2017008)

***Abstract*— This document is complete design and analysis of algorithm LU decompose of given square matrix.**

I. INTRODUCTION

For this problem ,we have found upper triangular part by row reducing input matrix. The lower triangular part is calculated by multiplying given matrix and inverse of upper triangular part.

This report further contains::
II. Algorithm Design.
III.Algorithm Analysis
IV.Experimental Study.
V.Conclusions.
VI.References.

II. ALGORITHM DESIGN

A. First Algorithm

We have found upper triangular part by row reducing input matrix. The lower triangular part is calculated by multiplying given matrix and inverse of upper triangular part.We use recursive approach of finding the determinant of matrices, where cofactor matrices are made and multiplied to the corresponding row elements and summation with proper sign is taken .Adjoint is calculated by taking the transpose of co-factor matrices. Upper triangularisation is done by row reducing the given matrix.

Algorithm 1

```
int main()
input:←n,A[m][m]
int A[M][M],U[M][M]
float L[M][M]←0
U←A
// row reducing matrix U
U←upper_triangular(U,n)
// To store adjoint of A[][]
int adj[M][M]
// To store inverse of A[][]
float inv[M][M]
adj←adjoint(U)
inv←inverse(U)
L←matmul(A,inv)
print←L
print←U
```

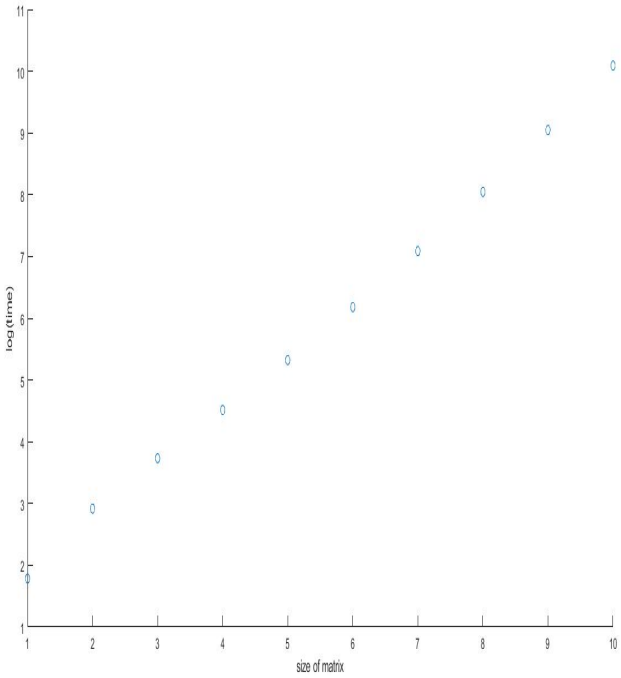
III. ALGORITHM ANALYSIS

A. Time Complexity

- The time complexities of different functions are given below.
  - get_co-factor:O(n^2)
  - Determinant:O(n^3)
  - Adjoint:O(n^7)
  - Inverse:O(n^7)
- The overall time complexity was calculated as 2*(O(n^3)+O(n^7)), i.e. O(n^7).

B. Space Complexity

- The space complexity was found to be 5*O(n^2),i.e. O(n^2)



IV. EXPERIMENTAL STUDY

The graph depicts, our calculation, that is the algorithm takes O(n^7) time is complied.

The integrated Development environment of C++ is used for processing the algorithm and graphical analysis is done using MATLAB scatter function.

V. CONCLUSIONS

In this document we conclude that our algorithm has time complexity equal to O(n^7) and the space complexity is of O(n^2).

VI. REFERENCES

- https://www.geeksforgeeks.org/finding-inverse-of-a-matrix-using-gauss-jordan-method/
- https://www.geeksforgeeks.org/determinant-of-a-matrix/
- https://www.geeksforgeeks.org/adjoint-inverse-matrix/