1. Time complexity of multiplying two matrices

Code:

#include <stdio.h>

#define N 3

void multiplyMatrix(int mat1[][N], int mat2[][N], int result[][N]) {

int i, j, k;

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

result[i][j] = 0;

for (k = 0; k < N; k++) {

result[i][j] += mat1[i][k] \* mat2[k][j];

}

}

}

}

void displayMatrix(int mat[][N]) {

int i, j;

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

printf("%d ", mat[i][j]);

}

printf("\n");

}

}

int main() {

int mat1[N][N], mat2[N][N], result[N][N];

int i, j;

printf("Enter elements of matrix 1 (%d x %d):\n", N, N);

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

scanf("%d", &mat1[i][j]);

}

}

printf("Enter elements of matrix 2 (%d x %d):\n", N, N);

for (i = 0; i < N; i++) {

for (j = 0; j < N; j++) {

scanf("%d", &mat2[i][j]);

}

}

multiplyMatrix(mat1, mat2, result);

printf("\nMatrix 1:\n");

displayMatrix(mat1);

printf("\nMatrix 2:\n");

displayMatrix(mat2);

printf("\nResultant Matrix:\n");

displayMatrix(result);

return 0;

}

Output:

Enter elements of matrix 1 (3 x 3):

1 2 3 4 5 6 7 8 9

Enter elements of matrix 2 (3 x 3):

9 8 7 6 5 4 3 2 1

Matrix 1:

1 2 3

4 5 6

7 8 9

Matrix 2:

9 8 7

6 5 4

3 2 1

Resultant Matrix:

30 24 18

84 69 54

138 114 90

--------------------------------

Process exited after 11.57 seconds with return value 0

Press any key to continue . . .

