Experiment - 1

Aim :- To implement Matrix Multiplication and analyse its time complexity.

Software Used :- Turbo C++.

Source Code :-

Program	Inst.	No. of times Executed
#include<iostream.h>		
#include< conio.h >		
Void main()		
{		
int A[3][3], B[3][3], C[3][3], i, j, k;	C_1	1
clrscr();	C_2	1
cout<<"Enter elements of first matrix : ";	C_3	1
for(i=0; i<3; i++)	C_4	$(n+1)$
{ for(j=0; j<3; j++)	C_5	$n(n+1)$
{ cin >> A[i][j];	C_6	n^2
}		
}		
cout<<"Enter elements of second matrix : ";	C_7	1
for(i=0; i<3; i++)	C_8	$(n+1)$
{ for(j=0; j<3; j++)	C_9	$n(n+1)$
{ cin>> B[i][j];	C_{10}	n^2
}		
}		

Program	Inst.	No. of times Executed
`for(i=0; i<3; i++)`	C_{11}	$(n+1)$
`{    for(j=0; j<3; j++)`	C_{12}	$n(n+1)$
`        {`		
`        C[i][j]=0;`	C_{13}	n^2
`        for(k=0; k<3; k++)`	C_{14}	$(n+1)n^2$
`          {`		
`          C[i][j]+= A[i][k] * B[K][j];`	C_{15}	n^3
`          }`		
`        }`		
`}`		
`cout<<"\n The Product of matrices is :-\n";`	C_{16}	1
`for(i=0; i<3; i++)`	C_{17}	$(n+1)$
`{    for(j=0; j<3; j++)`	C_{18}	$n(n+1)$
`        {`		
`          cout << C[i][j] << " ";`	C_{19}	n^2
`        }`		
`      cout<<endl;`	C_{20}	n
`  }`		
`getch();`	C_{21}	1
`}`		

Total time, $t(n) = C_1 + C_2 + C_3 + C_4(n+1) + C_5(n+1)n + C_6 n^2 + C_7 + C_8(n+1) + C_9(n+1)n + C_{10}.n^2 + C_{11}(n+1) + C_{12}.n(n+1) + C_{13} n^2 + C_{14}.n^2(n+1) + C_{15} n^3 + C_{16} + C_{17}(n+1) + C_{18} n(n+1) + C_{19} n^2 + C_{20}.n + C_{21}$

$$\begin{aligned}\Rightarrow t(n) = {} & (C_{14} + C_{15}) n^3 + (C_5 + C_6 + C_9 + C_{10} + C_{12} \\ & + C_{13} + C_{14} + C_{18} + C_{19}) n^2 + (C_4 + C_5 \\ & + C_8 + C_9 + C_{11} + C_{12} + C_{17} + C_{18} + C_{20}) n \\ & + (C_1 + C_2 + C_3 + C_4 + C_7 + C_8 + C_{11} + C_{16} + C_{17} \\ & + C_{21})\end{aligned}$$

$$\Rightarrow t(n) = An^3 + Bn^2 + Cn + D$$

$$\Rightarrow t(n) = O(n^3)$$

$\therefore$ Time Complexity of the program is $O(n^3)$.