```c
//Tower of Hanoi
#include<stdio.h>
#include<time.h>
#include<math.h>
void towers(int n,char s, char t, char d)
{
    if(n==1){
        printf("Move disk 1 from %c to %c\n",s,d);
        return;
    }
    towers(n-1,s,d,t);
    printf("Move disk %d from %c to %c\n",n,s,d);
    towers(n-1,t,s,d);
}
int main()
{
    int n;
    printf("Enter no. of disks : ");
    scanf("%d",&n);
    double toh_t
    clock_t begin=clock();
    towers(n,'S','T','D');
    printf("\nTotal Steps : %lf",(pow(2,n)-1));
    clock_t end=clock();
    toh_time+=(double)(end-begin)/CLOCKS_PER_SEC;
    printf("\nn=%d\tTime:%f\n",n,toh_time);
    return 0;
}
```

int main::n

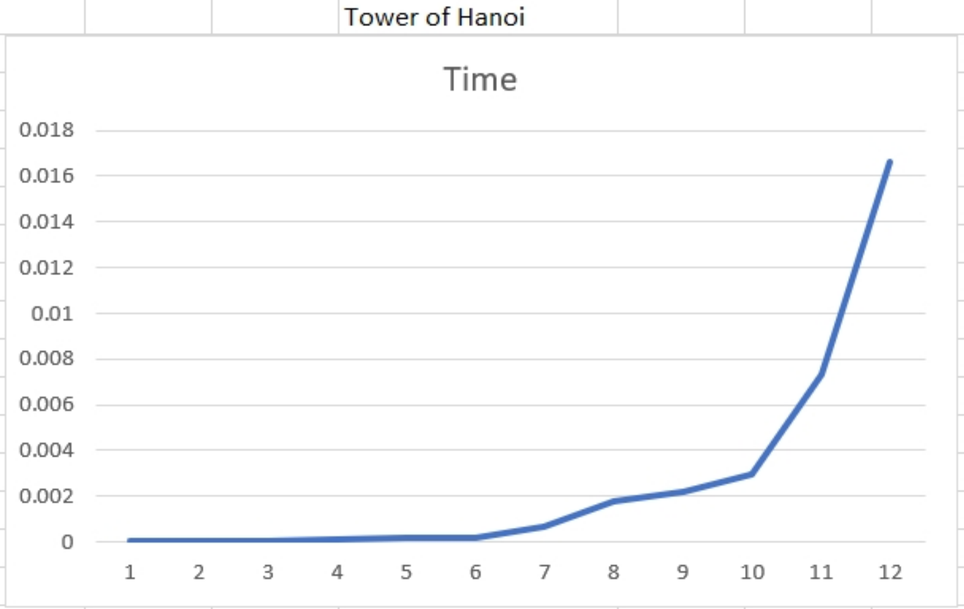
"C:\Users\Shreshtha Aggarwal\Desktop\4sem\bin\Debug\4sem.exe"

```
Enter no. of disks : 5
Move disk 1 from S to D
Move disk 2 from S to T
Move disk 1 from D to T
Move disk 3 from S to D
Move disk 1 from T to S
Move disk 2 from T to D
Move disk 1 from S to D
Move disk 4 from S to T
Move disk 1 from D to T
Move disk 2 from D to S
Move disk 1 from T to S
Move disk 3 from D to T
Move disk 1 from S to D
Move disk 2 from S to T
Move disk 1 from D to T
Move disk 5 from S to D
Move disk 1 from T to S
Move disk 2 from T to D
Move disk 1 from S to D
Move disk 3 from T to S
Move disk 1 from D to T
Move disk 2 from D to S
Move disk 1 from T to S
Move disk 4 from T to D
Move disk 1 from S to D
Move disk 2 from S to T
Move disk 1 from D to T
Move disk 3 from S to D
Move disk 1 from T to S
Move disk 2 from T to D
Move disk 1 from S to D

Total Steps : 31.000000
n=5     Time:0.015000

Process returned 0 (0x0)   execution time : 2.531 s
Press any key to continue.
```

Disks	Time
1	0.000031
2	0.000046
3	0.000079
4	0.000149
5	0.000179
6	0.000166
7	0.000692
8	0.001766
9	0.002185
10	0.002969
11	0.007319
12	0.016592



```c
//DFS Traversal
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int G[10][10],v[10],n,a[1][10];
void dfs(int i)
{
    int j;
    printf("\n%d",i);
    v[i]=1;
    for(j=0;j<n;j++){
       if(!v[j]&&G[i][j]==1)
            dfs(j);
    }
}
void dfs_c(int n,int G[10][10],int m,int s[])
{
    int y;
    s[m]=1;
    for(y=0;y<n;y++){
        if((G[m][y]==1)&&(!s[y]))
            dfs_c(n,G,y,s);
    }
}
int main()
{
    int i,j,con,s[10],flag;
    printf("Enter number of vertices : ");
    scanf("%d",&n);
    printf("\nEnter adjecency matrix of the graph :\n");
    for(i=0;i<n;i++)
    {
        printf("Enter row %d : -\n",i+1);
```

```
416         for(j=0;j<n;j++)
417             scanf("%d",&G[i][j]);
418      }
419     for(i=0;i<n;i++)
420         v[i]=0;
421     printf("DFS Traversal order :-\n");
422     double dfs_time=0.0;
423     clock_t begin=clock();
424     dfs(0);
425     con=0;
426     for(j=0;j<n;j++){
427         for(i=0;i<n;i++)
428             s[i]=0;
429         dfs_c(n,G,j,s);
430         flag=0;
431         for(i=0;i<n;i++){
432             if(s[i]==0)
433                 flag=1;
434         }
435         if(flag==0)
436             con=1;
437     }
438     if(con==1)
439        printf("\nGraph is connected\n");
440      else
441        printf("\nGraph is not connected\n");
442     clock_t end=clock();
443     dfs_time+=(double)(end-begin)/CLOCKS_PER_SEC;
444     printf("\nn=%d\tTime:%f\n",n,dfs_time);
445     return 0;
446 }
447
```

```
Enter number of vertices : 3

Enter adjecency matrix of the graph :
Enter row 1 : -
0
1
0
Enter row 2 : -
0
0
1
Enter row 3 : -
1
0
0
DFS Traversal order :-

0
1
2
Graph is connected

n=3     Time:0.000038


...Program finished with exit code 0
Press ENTER to exit console.
```

DFS Traversal

No. of Vertices	Time
0	0.000014
1	0.000015
2	0.00002
3	0.000015
4	0.000018
5	0.000022
6	0.000056
7	0.000063