

# CENTRAL UNIVERSITY OF HARYANA

Department of Computer Science & Engineering under SOET



## ANALYSIS AND DESIGN OF ALGORITHM LAB

WAP to implement a Fibonacci series and plots its run time graph.

Submitted by  
**Ajit singh**  
Roll No- 192219

Submitted to  
**Dr. Manish Kumar**  
Assistant Professor  
Central University of Haryana (SOET)

**Assignment-1 :** Write a program to implement a Fibonacci series and plots its run time for 10, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 terms.

**For Loop Approach-:** In case of for loop approach we will solve the program using for loop and we will save the previous two elements such that we don't have to find the previous to element again and again. Time complexity comes out to be  $O(n)$ .

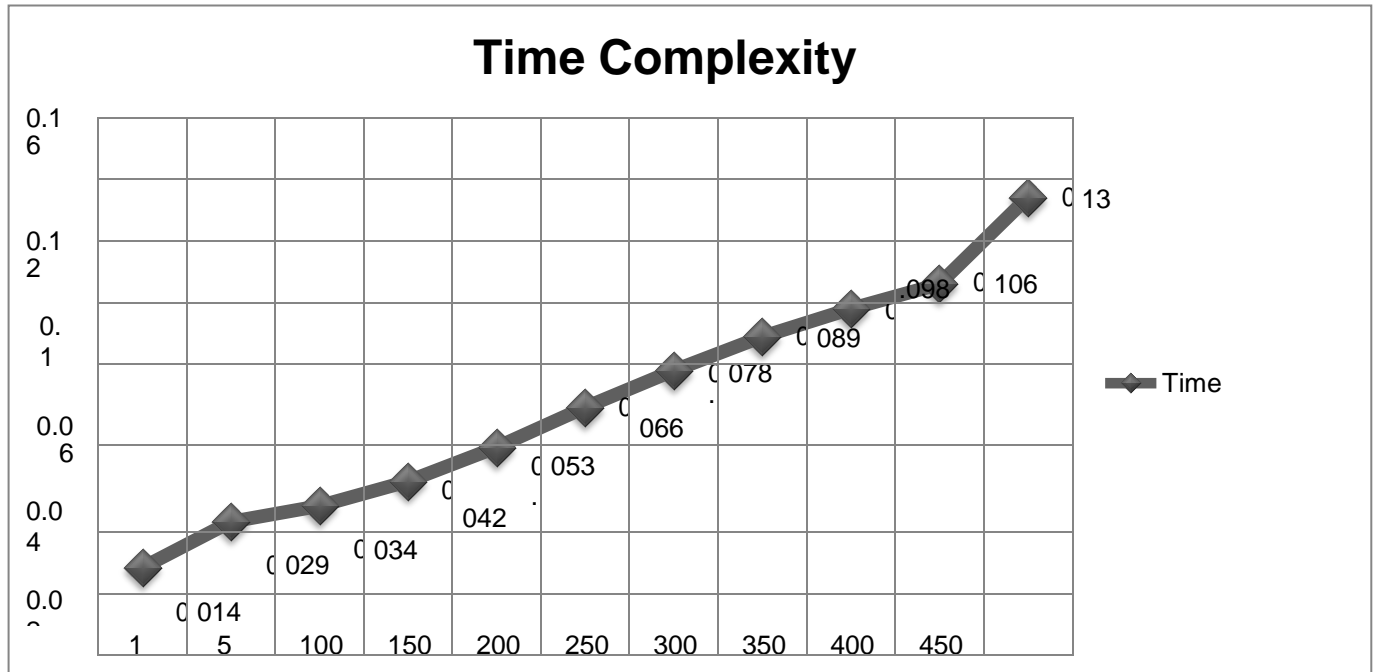
**Code:**

```
#include <iostream>
using namespace std;
int main()
{
    double a, b, c, d;
    cout << "Enter the number of elements: ";
    cin >> d;
    a = 0;
    b = 1;
    cout << a << " " << b << " ";
    for (int i = 1; i < d; i++) //Using For Loop To Find Fibonnaci Series
    {
        c = a + b;
        a = b;
        b = c;
        cout << c << " ";
    }
    return 0;
}
```

**Observation Table :**

No Of Inputs	Average Run Time (in seconds)
10	0.014
50	0.029
100	0.034
150	0.042
200	0.053
250	0.066
300	0.078
350	0.089
400	0.098
450	0.106
500	0.134

**Graph :**



**Recursion Approach:-** In Recursion Approach We will find the Fibonacci series elements using recursion and finding the element at a particular position. The Time complexity in Recursion is  $O(n^2)$  which means the time complexity will be more in case of recursion approach for small number of elements as well.

**Code:-**

```
#include<iostream>
using namespace std;
double fibonacci(int n)
{
    if(n==0)
    {
        return 0;
    }
    else if(n==1)
    {
        return 1;
    }
    else
    return fibonacci(n-1)+fibonacci(n-2);
}
```

ADA

```

int main()
{
    double a;
    cout<<"Enter the Number of Element :";
    cin>>a;
    for(int i=0;i<a;i++)
    {
        cout<<fibonacci(i)<<" ";
    }

    return 0;
}

```

**Observation Table:-**

No of Terms	Average Run Time (in seconds)
10	0.02885
20	0.03545
30	0.05985
40	3.029
50	299.1

**Graph:-**

