Experiment 1

WAP to perform the empirical analysis of iterative algorithm to find nth Fibonacci number.

Program: -

```
#include<stdio.h>
#include<time.h>
int main()
{
   int n, i;
   double first=0, second=1, temp, time;
   clock_t start, end;
   printf("Enter the position of fibonacci number:");
   scanf("%d", &n);
   start = clock();
   printf("%f, %f", first, second);
   i = 3;
   while(i <= n)
   {
      temp = first + second;
      first = second;
      second = temp;
      printf("%f \n", temp);
      i++;
   }
   printf("\n\n\n");
   end = clock();
   printf("The nth fibonacci number is: %lf \n",temp);
   time = ((double)(end-start)*1000) / CLOCKS_PER_SEC;
   printf("Time=%lf miliseconds", time);
}
```

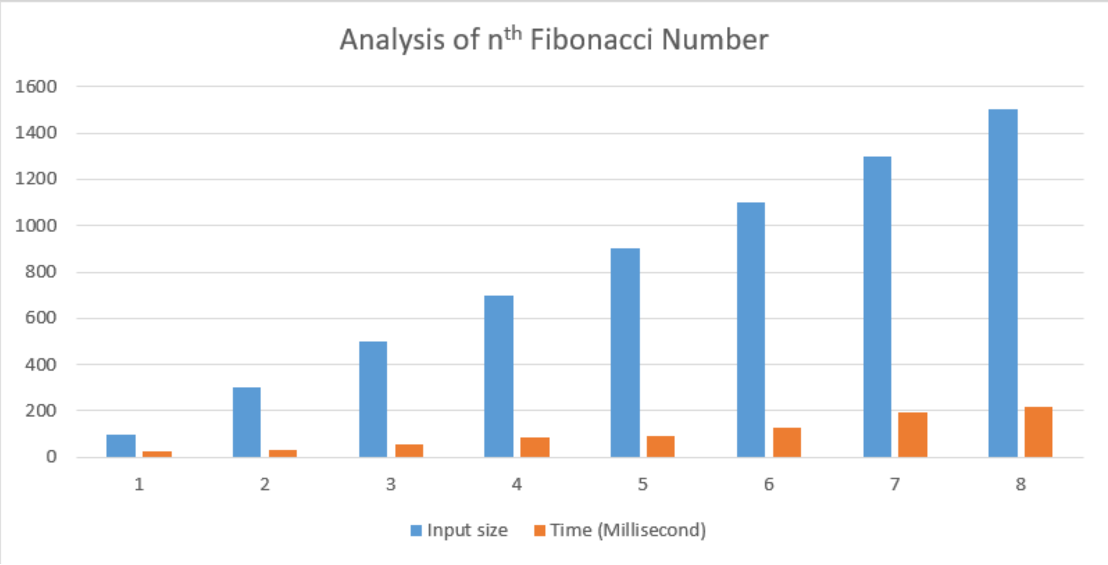
Result Analysis and Discussion:

This experiment has been conducted in a 64-bit system with 16 GB RAM and Processor 12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz. The algorithm is implemented in C programming language in Visual Studio Code 1.81.1 Code Editor.

In this experiment the algorithm to find the n^{th} Fibonacci number has been implemented and executed for different value of n. During this experiment for different value of n the time taken by the algorithm has been measured and tabulated as shown in table below.

Input Size	**Time (Milliseconds)**
100	26
300	31
500	57
700	84
900	93
1100	125
1300	194
1500	216

The graph shown below is the plot of input n and the time in milliseconds taken by the algorithm while running on a system recorded in table above.



Based on the above table and graph it is clearly seen that the input number n has linear relationship with the time taken by the system to find the nth Fibonacci number of input number n.

Conclusion:

In this experiment it has been found that the size of input (n) has linear relationship with the time taken by the system to find the n^{th} Fibonacci number. This is equivalent with the asymptotic time complexity of the algorithm. Hence, this experiment proves complexity of the algorithm to find n^{th} Fibonacci number is O (n).

Experiment 2

WAP to perform the empirical analysis of iterative algorithm for linear search.

Program: -

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int LinearSearch(int *arr, int n,int key);
int main()
{
   clock_t start, end;
   double time;
   int arr[500000],n,key, num;
   printf("-----Linear Searching--------\n");
   printf("\nEnter size of array\n");
   scanf("%d",&n);
   printf("%d numbers\n",n);
   for (int i = 0; i < n; i++)
   {
      num = (rand()%10000);
      arr[i] = num;
   }
   printf("Enter the key\n");
   scanf("%d",&key);
   start = clock();
   int ans = LinearSearch(arr,n,key);
   if(ans!=-1)
   {
      printf("\n%d is in %d index.\n",key,ans);
   }
   else
   {
      printf("\nNot Found\n");
```

```
        }
        end = clock();
        time=((double)(end-start)*10000) / CLOCKS_PER_SEC;
        printf("Time=%lf microseconds", time);
    }

    int LinearSearch(int *arr, int n,int key)
    {
        for(int i=0;i<n; i++)
        {
            if(arr[i]== key)
            {
                return i+1;
            }
        }
        return -1;
    }
```

Result Analysis and Discussion:

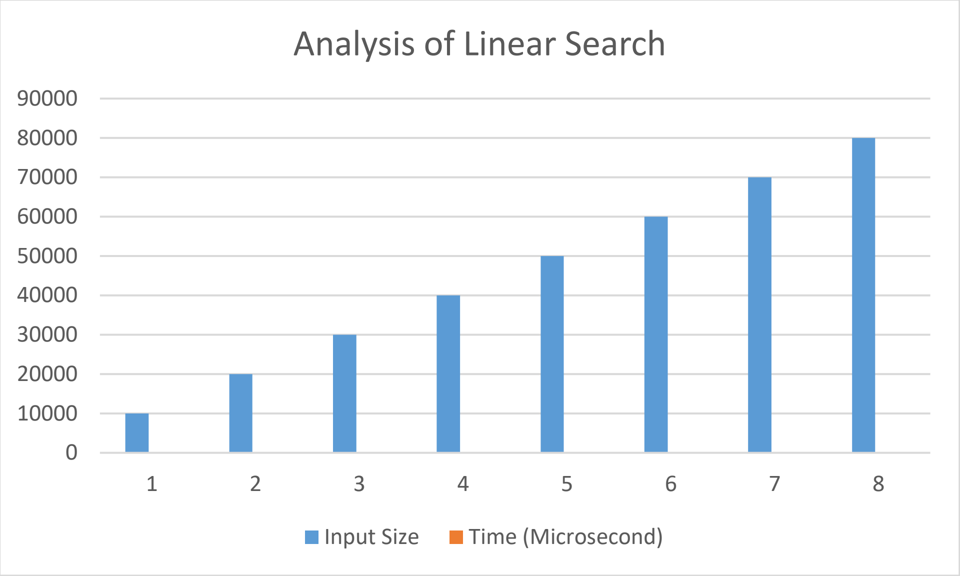
This experiment has been conducted in a 64-bit system with 16 GB RAM and Processor 12th Gen Intel(R) Core (TM) i5-12500H 3.10 GHz. The algorithm is implemented in C programming language in Visual Studio Code 1.81.1 Code Editor.

In this experiment the algorithm to search the given key has been in the array by using linear search. During this experiment for different n size of array and different key is taken by the algorithm has been measured and tabulated as shown in table below.

Input Size	**Time (Microsecond)**
10000	20
20000	20
30000	30
40000	10
50000	20
60000	30
70000	20

80000	20

The graph shown below is the plot of input n and the time in microsecond taken by the algorithm while running on a system recorded in table above.



Based on the above table and graph it is clearly seen that the size of array n has linear relationship with the time taken by the system to find the key in it.

Conclusion:

In this experiment it has been found that the size of array(n) has linear relationship with the time taken by the system to find the key in it. This is equivalent with the asymptotic time complexity of the algorithm. Hence, this experiment proves complexity of the algorithm to linear search is O (n).