1.1.1

-----------------------Recursion------------------------------------------

**void** printFibonacci(**int** n){

**static** **int** n\_1=0, n\_2=1, n\_3;

**if**(n>0){

         n\_3 = n\_1 + n\_2;

         n\_1 = n\_2;

         n\_2 = n\_3;

 cout<<n\_3<<" ";

         printFibonacci(n-1);

    }

}

**int** main(){

**int** n;

    cout<<"Enter the number of elements: ";

    cin>>n;

    cout<<"Fibonacci Series: ";

    cout<<"0 "<<"1 ";

    printFibonacci(n-2);  //n-2 because 2 numbers are already printed

**return** 0;  }

1.1.2

Complexity is T(n) = T(n-1) + T(n-2)

1.1.3

------------------ Without Recursion--------------------------------------

**int** main() {

**int** n\_1=0,n\_2=1,n\_3,i,number;

 cout<<"Enter the number of elements: ";

 cin>>number;

 cout<<n\_1<<" "<<n\_2<<" "; //printing 0 and 1

**for**(i=2;i<number;++i) //loop starts from 2

 {

  n\_3=n\_1+n\_2;

  cout<<n\_3<<" ";

  n\_1=n\_2;

  n\_2=n\_3;

 }

**return** 0;

   }

1.1.4 Complexity is O(n)

2.1

-------Python------------------------

from random import \*

from math import sqrt

inside=0

n=1000

for i in range(0,n):

x=random()

y=random()

if sqrt(x\*x+y\*y)<=1:

inside+=1

pi=4\*inside/n

print pi

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

I am sorry that something is wrong with my git and I cannot use “All About Pull Request”.

Thanks, Haixia Guo (Grace) 2/3/2019.