from numpy import zeros

from time import time

# fibonacci function with 3 variables

def fibonacci\_3\_variables(num):

    if (num < 3):

        return 1

    a, b, c = 1, 1, 0

    for i in range(num-2):

        c = a + b

        a,b = b,c

    return c

# fibonacci function with 2 variables

def fibonacci\_2\_variables(num):

    if (num < 3):

        return 1

    a, b = 1, 1

    for i in range(num-2):

        b += a

        a = b - a

    return b

# recursive fibonacci function

def recursive\_fibonacci(num):

    if (num <= 2):

        return 1

    return recursive\_fibonacci(num-1) + recursive\_fibonacci(num-2)

# fibonacci function with array

def fibonacci\_with\_array(num):

    if (num <= 2):

        return 1

    array = zeros(num)

    array[0], array[1] = 1, 1

    for i in range(num-2):

        array[i+2] = array[i+1] + array[i]

    return int(array[i+2])

fibonacci\_3\_variables\_time, fibonacci\_2\_variables\_time, recursive\_fibonacci\_time, fibonacci\_with\_array\_time = 0, 0, 0, 0

input\_data = open('D:/Education/Data Structure/Projects/Fibonacci/InputData.txt','r')

output\_data = open('D:/Education/Data Structure/Projects/Fibonacci/OutputData.txt','w')

for i in range(20):

    num = int(input\_data.readline())

    start = time()

    fibonacci\_3\_variables(num)

    end = time()

    fibonacci\_3\_variables\_time += end - start

    output\_data.write('fibonacci function with 3 variables: f( %d ) = %d , time = %.20f '  % (num, fibonacci\_3\_variables(num), end - start) + '\n' )

    print('fibonacci function with 3 variables: f( %d ) = %d , time = %.20f '  % (num, fibonacci\_3\_variables(num), end - start) + '\n' )

    start = time()

    fibonacci\_2\_variables(num)

    end = time()

    fibonacci\_2\_variables\_time += end - start

    output\_data.write('fibonacci function with 2 variables: f( %d ) = %d , time = %.20f  '  % (num, fibonacci\_2\_variables(num), end - start) + '\n' )

    print('fibonacci function with 2 variables: f( %d ) = %d , time = %.20f  '  % (num, fibonacci\_2\_variables(num), end - start) + '\n' )

    start = time()

    recursive\_fibonacci(num)

    end = time()

    recursive\_fibonacci\_time += end - start

    output\_data.write('recursive fibonacci function : f( %d ) = %d , time = %.20f  '  % (num, recursive\_fibonacci(num), end - start) + '\n' )

    print('recursive fibonacci function : f( %d ) = %d , time = %.20f  '  % (num, recursive\_fibonacci(num), end - start) + '\n' )

    start = time()

    fibonacci\_with\_array(num)

    end = time()

    fibonacci\_with\_array\_time += end - start

    output\_data.write('fibonacci function with array : f( %d ) = %d , time = %.20f  '  % (num, fibonacci\_with\_array(num), end - start) + '\n'\*2 )

    print('fibonacci function with array : f( %d ) = %d , time = %.20f  '  % (num, fibonacci\_with\_array(num), end - start) + '\n'\*2 )

output\_data.write( "fibonacci function with 3 variables total time : %.20f \nfibonacci function with 2 variables total time : %.20f \nrecursive fibonacci function total time : %.20f\nfibonacci function with array total time : %.20f " % (fibonacci\_3\_variables\_time , fibonacci\_2\_variables\_time, recursive\_fibonacci\_time, fibonacci\_with\_array\_time))

print("fibonacci function with 3 variables total time : %.20f \nfibonacci function with 2 variables total time : %.20f \nrecursive fibonacci function total time : %.20f\nfibonacci function with array total time : %.20f \n" % (fibonacci\_3\_variables\_time , fibonacci\_2\_variables\_time, recursive\_fibonacci\_time, fibonacci\_with\_array\_time))

input\_data.close()

output\_data.close()