Mahendra Chaurasiya

18BEC026

ASSIGNMENT 1

INTRODUCTION TO ALGORITHMS

# Consider the following Fibonacci series and solve the following conditions

fib (n) = fib(0), fib (1), fib (2),………..fib(n) where fib(n) = fib(n-1) + fib(n-2)

**a) Draw the Flowchart , Algorithms in pseudocode for solving .**

# Pseudocode for fibonacci series of n number

**Step1 :-START**

**Step2 :-Read n from user**

**Step3 :-define the function fib**

**Step4 :-Initialize integer variables a=0 & b=1**

**Step5 :-Check value of n**

**Step6 :-less than 0 then go to step 10**

**Step7 :-equal to 0 then print a**

**Step8 :-greater than or equal to 1**

**Step9 :-for 1<=i<=n**

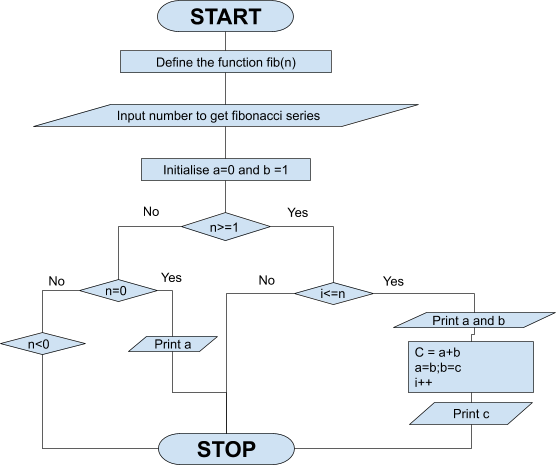
**C = a +b Print c**

**a=b;b=c**

**I = i + 1**

**Step10 :-STOP**

**FLOW CHART:**



## b)Write two types of algorithm ( recursive and non recursive ) for fib(5) and fib(500) series

Recursive Algorithm for fib(5)

Step-1 : Start

Step-2 : declare variables a, b, c, i and n

Step-3 : Initialize a=0, b=1, i=2 and n=5

Step-4 : Print a and b

Step-5 : if(i>n) then go to step 12

Step-6 : c = a+b

Step-7 : Print c

Step-8 : a=b

Step-9 : b=c

Step-10 : i=i+1

Step-11 : Go to step-5

Step-12 : Stop.

Recursive Algorithm for fib(500)

Step-1 : Start

Step-2 : declare variables a, b, c, i and n

Step-3 : Initialize a=0, b=1, i=2 and n=500

Step-4 : Print a and b

Step-5 : if(i>n) then go to step 12

Step-6 : c = a+b

Step-7 : Print c

Step-8 : a=b

Step-9 : b=c

Step-10 : i=i+1

Step-11 : Go to step-5 Step-12 : Stop.

Iterative Algorithm for fib(5)

Step1 :-START

Step2 :-Read n = 5

Step3 :-define the function fib

Step4 :-Initialize integer variables a=0 & b=1

Step5 :-Check value of n

Step6 :-less than 0 then go to step 10

Step7 :-equal to 0 then print a

Step8 :-greater than or equal to 1

Step9 :-for 1<=i<=5

C = a +b

Print c

a=b;b=c

I = i + 1

Step10 :-STOP

Iterative Algorithm for fib(500)

Step1 :-START

Step2 :-Read n = 500

Step3 :-define the function fib

Step4 :-Initialize integer variables a=0 & b=1

Step5 :-Check value of n

Step6 :-less than 0 then go to step 10

Step7 :-equal to 0 then print a

Step8 :-greater than or equal to 1

Step9 :-for 1<=i<=500

C = a +b Print c a=b;b=c

I = i + 1

Step10 :-STOP

## c) Find out the Total memory or space required to perform these​ Fibonacci series computational operations

For Iterative method :​

Total Memory = 4 Bytes \* 5 variables = 20 Bytes Therefore , Space complexity is O(1).

For Recursion method :​

Total Memory = 4 Bytes \* 5 variables + O(n) = 20 Bytes + O(n) = O(n) Therefore , Space complexity is O(n) .

## d) Find out the WORST CASE and BEST CASE scenario from the​ above identified approaches

Recursive Fibonacci Algorithm holds the worst case scenario , as it occupies O(n) space , the total memory consumption depends on the n . Iterative Fibonacci Algorithm holds the best case scenario , as it occupies O(1) space , the total memory consumption doesn’t depend on the n.

## e) Write a program and compare the actual memory consumed by all the approaches

Iterative method:

|  |
| --- |
| import​ os |
| import​ psutil |
|  |
| def​ ​fib​(​n​): |
|  |
| a = ​0 |
| b = ​1 |
|  |
| ​if​ n < ​0​: |
| ​print​(​"number entered is not valid"​) |
|  |
| ​elif​ n==​0​: |
| ​print​(a) |
|  |
| ​elif​ n>=​1​: |
| ​print​(a) |
| ​print​(b) |
|  |
| ​for​ i ​in​ ​range​(​1​,n): |
| c = a + b |
| a = b |
| b = c |
| ​print​(c) |
|  |
|  |
| import​ time |
|  |
| start = time.time() |
|  |
| n = ​int​(​input​(​"enter the n.o of fibonacci to be generated:"​)) |
| fib(n) |
|  |
| process = psutil.Process(os.getpid()) |
| print​(​f​"Total memory : ​{​process.memory\_info().rss​}​"​) |
|  |
| end = time.time() |
| print​(​f​"time taken: ​{​end - start​}​"​) |
|  |
|  |
|  |
|  |

Recursive method:

|  |
| --- |
| import​ os |
| import psutil​ |
| import​ time |
| def​ ​fib​(​n​): |
|  |
| ​if​ n < ​0​: |
| ​print​(​"number entered is not valid"​) |
|  |
| ​elif​ n == ​0​: |
| ​return​ ​0 |
|  |
| ​elif​ n == ​1​: |
| ​return​ ​1 |
| ​else​: |
| ​return​ fib(n-​1​)+fib(n-​2​) |
|  |
| start = time.time() |
|  |
| k = ​int​(​input​(​"enter Fibonacci sequence index number: "​)) |
| print(​ fib(k)) |
|  |
| process = psutil.Process(os.getpid()) |
| print(​ process.memory\_info().rss) |
|  |
| end = time.time() |
| print​(​f​"time taken: ​{​end - start​}​"​) |

## Checking that given number is in fibonacci series

|  |
| --- |
|  |
| **import**​ **math** |
| **import**​ **os** |
| **import**​ **psutil** |
|  |
| **def**​​**isPerfectSquare**​**(**​**x**​**):** |
| **s =** ​**int**​**(math.sqrt(x))** |
| ​**return**​ **s\*s == x** |
|  |
|  |
| **def**​​**isFibonacci**​**(**​**n**​**):** |
|  |
| ​**return**​ **isPerfectSquare(**​**5**​**\*n\*n +** ​**4**​**)** ​**or**​ **isPerfectSquare(**​**5**​**\*n\*n -** ​**4**​**)** |
|  |
| **n =** ​**int**​**(**​**input**​**(**​**"enter the n.o to check :"**​**))** |
| **if**​ **(isFibonacci(n) ==** ​**True**​**):** |
| ​**print**​ **(n,**​**"is a Fibonacci Number"**​**)** |
| **else**​**:** |
| ​**print**​ **(n,**​**"is a not Fibonacci Number "**​**)** |
|  |
| **process = psutil.Process(os.getpid())** |
| **print**​**(**​**f**​**"Total memory :** ​**{**​**process.memory\_info().rss**​**}**​**"**​**)** |