Fibonacci Series

The Fibonacci sequence was invented by the Italian Leonardo Pisano Bigollo (1180 − 1250), who is known in mathematical history by several names: Leonardo of Pisa (Pisano means from Pisa) and Fibonacci (which means "son of Bonacci").

While growing up in North Africa, Fibonacci learned the more efficient Hind − Arabic system of arithmetical notation (1, 2, 3, 4 …) from an Arab teacher. In 1202, he published his knowledge in a famous book called the Liber Abaci (which means the "book of the abacus," even though it had nothing to do with the abacus).

The Liber Abaci showed how superior the Hindu − Arabic arithmetic system was to the Roman numeral system, and it showed how the Hindu − Arabic system of arithmetic could be applied to benefit Italian merchants.

The Fibonacci sequence was the outcome of a mathematical problem about rabbit breeding that was posed in the Liber Abaci. The problem was this: Beginning with a single pair of rabbits (one male and one female), how many pairs of rabbits will be born in a year, assuming that every month each male and female rabbit gives birth to a new pair of rabbits, and the new pair of rabbits itself starts giving birth to additional pairs of rabbits after the first month of their birth?

Months	*Newborns (can't reproduce)*	*One − month − olds (can't reproduce)*	*Mature Pairs (can reproduce)*	*Total Pairs*
1	**1**	**+0**	**+0**	**= 1**
2	**0**	**+1**	**+0**	**= 1**
3	**1**	**+0**	**+1**	**= 2**
4	**1**	**+1**	**+1**	**= 3**
5	**2**	**+1**	**+2**	**= 5**
6	**3**	**+2**	**+3**	**= 8**
7	**5**	**+3**	**+5**	**= 13**
8	**8**	**+5**	**+8**	**= 21**
9	**13**	**+8**	**+13**	**= 34**
10	**21**	**+13**	**+21**	**= 55**

Each number in the tablerepresents a pair of rabbits. Each pair of rabbits can only give birth after its first month of life. Beginning in the third month, the number in the Mature pairs *column represents the number of pairs that can bear rabbits.*

The numbers in the Total Pairs*column represent the Fibonacci sequence.*

That is : 1st month have 1 pair and 2nd month have 1 pair . Hence for `0` month , it will be `0` pairs.

This we represent it with a program:

```
int fib(int n) // Function to calculate the nth Fibonacci number
{
    int a = 0, b = 1, c; // Declare variables
    if (n == 0) // Base case
    {
        return a; // If n is 0, return a
    }
    for (int i = 1; i < n; i++) // Loop to calculate the nth Fibonacci number
    {
        c = a + b; // Calculate the sum of the previous two terms and store it in c
        a = b; // Assign the value of b to a
        b = c; // Assign the value of c to b
    }
    return b; // Return statement
}
```

```
//Function Call

for (int i = 0; i < n; i++) // Loop to print the first
n Fibonacci numbers
    {
        cout << fib(i) << " "; // Print the ith
Fibonacci number
    }
```

$When\ n = 1, fib(0)\ it\ will\ return\ a\ i.e.0.[Note: i < n].$

$When\ n = 2, fib(1)\ it\ will\ return\ b\ i.e.1\ .[Note: i < n].$

$When\ \ n = 3\ , fib\ (2)\ it\ will\ now\ enter\ the\ loop\ of\ the\ function.$

$c = a + b = \ 0 + 1 = 1.$

$a = b = 1.$

$b = c = 1.$

$Hence\ it\ will\ return\ b\ i.e.1.$

$Next\ when\ n = 4\ , we\ will\ get\ b = c = a + b = 1 + 1 = 2\ .$

$Hence\ the\ sequence\ will\ be : 0\ 1\ 1\ 2\ 3\ 5 \ldots etc.exactly\ the$

$\boldsymbol{from\ the\ table\ .}$

$\boldsymbol{i.e. Fibonacci\ sequence\ \Longrightarrow T_n = T_{n-1} + T_{n-2}\ , n > 1\ and}$
$\boldsymbol{T_0 = 0\ , when\ n = 0\ also, T_1 = 1, when\ n = 1\ (Base\ Cases).}$

$\boldsymbol{Now\ converting\ it\ to\ Recursion:}$

```
int fibonacci(int n) // Function to calculate the nth
Fibonacci number
{
    if (n == 0 || n == 1) // Base case
    {
        return n; // If n is 0 or 1, return n
    }
    return fibonacci(n-1) + fibonacci(n-2); //
Recursive call : nth Fibonacci number is the sum of
(n-1)th and (n-2)th Fibonacci number
}

//Function Call

 for (int i = 0; i < n; i++) // Loop to print the
first n Fibonacci numbers
    {
        cout << fibonacci(i) << " "; // Print the ith
Fibonacci number
    }
```

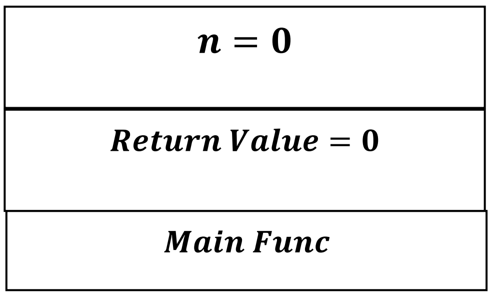
$Suppose\ n = 5$

$Hence\ \ i\ will\ go\ from\ 0\ to\ 4\ , then$:

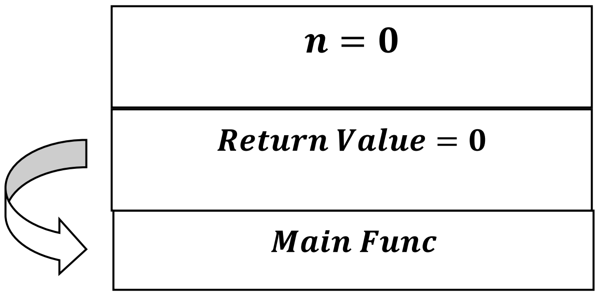
$fibonacci(0)\ , we\ get$:

$It\ return\ the\ base\ case\ `n`\ :\ `0`$

$1st\ Push$



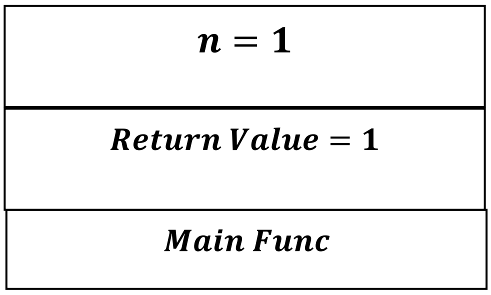
$1st\ Pop$



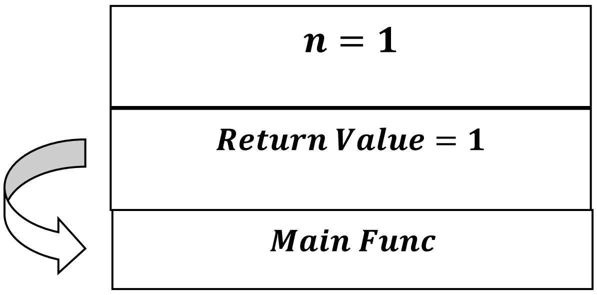
***fibonacci*(1) , *we get*:**

It return the base case `n` : `1`

2nd Push



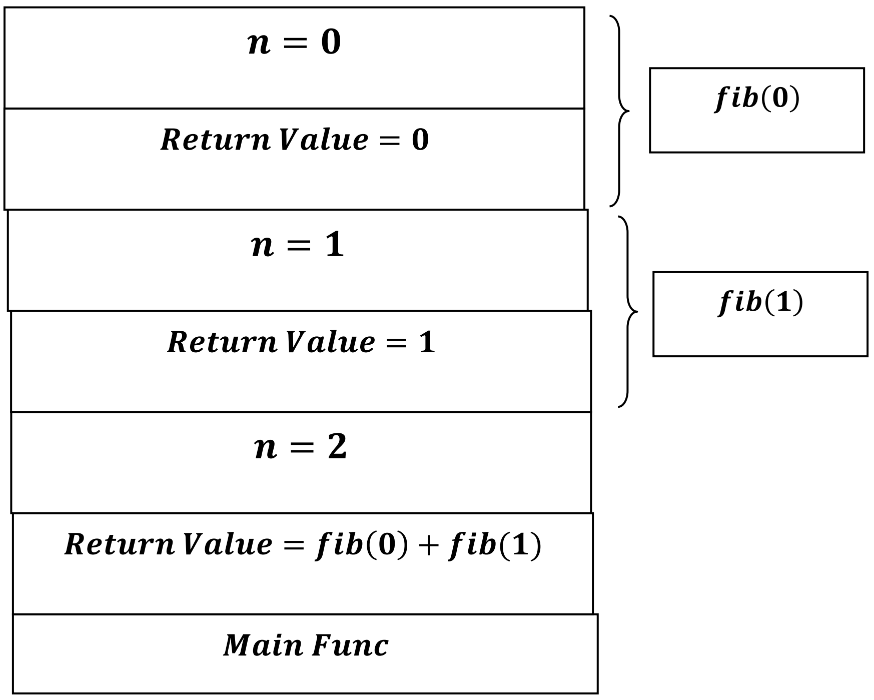
2nd Pop



***fibonacci*(2) , *we get*:**

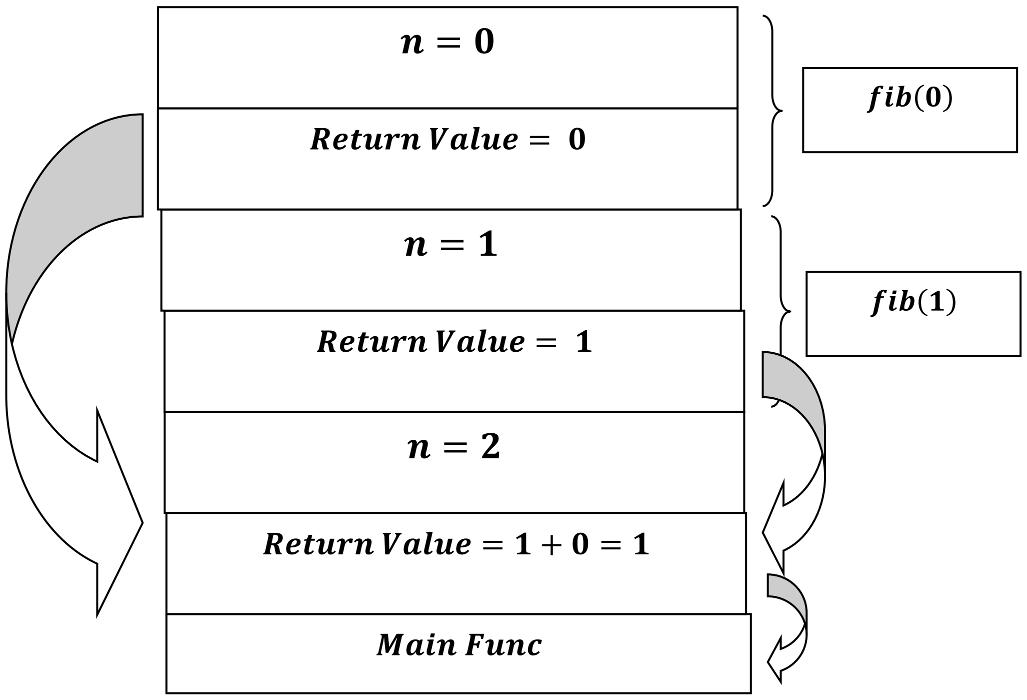
***As fibonacci*(2) *will return* : *fibonacci*(1) + *fibonacci*(0)**

$3rd\ Push$



$n = 0$
$Return\ Value = 0$
$n = 1$
$Return\ Value = 1$
$n = 2$
$Return\ Value = fib(0) + fib(1)$
$Main\ Func$
$fib(0)$
$fib(1)$

$3rd\ Pop$

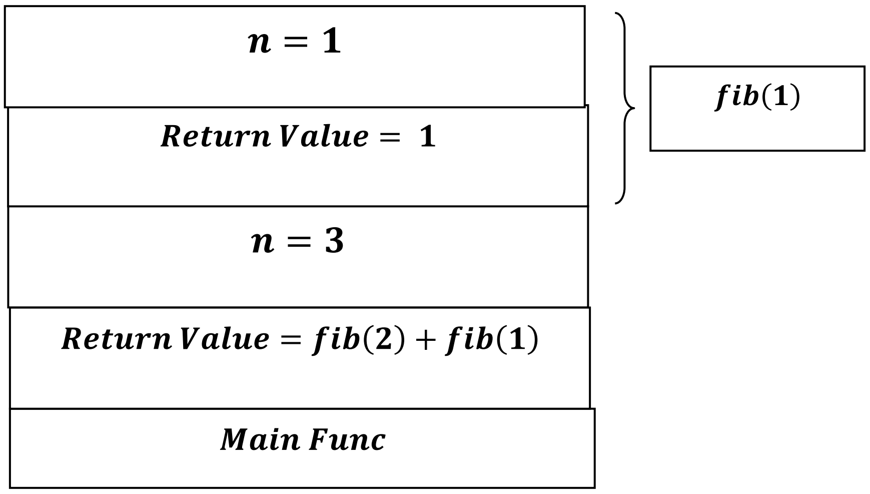


$Hence\ upto\ now\ we\ got\ series: 0\ , 1, 1$

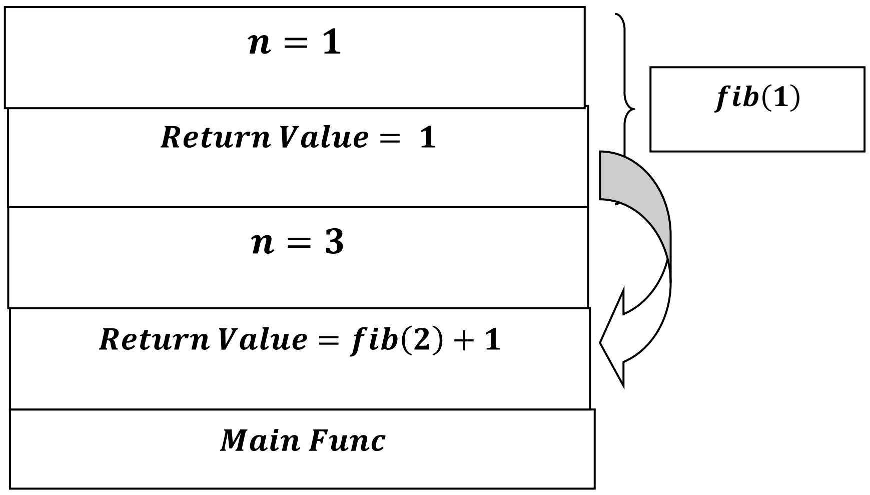
$fibonacci(3)\ , we\ get:$

$As\ fibonacci(3)\ will\ return:\ fibonacci(2) + \ fibonacci(1)$

$\mathbf{4.1.}\,\boldsymbol{push}$

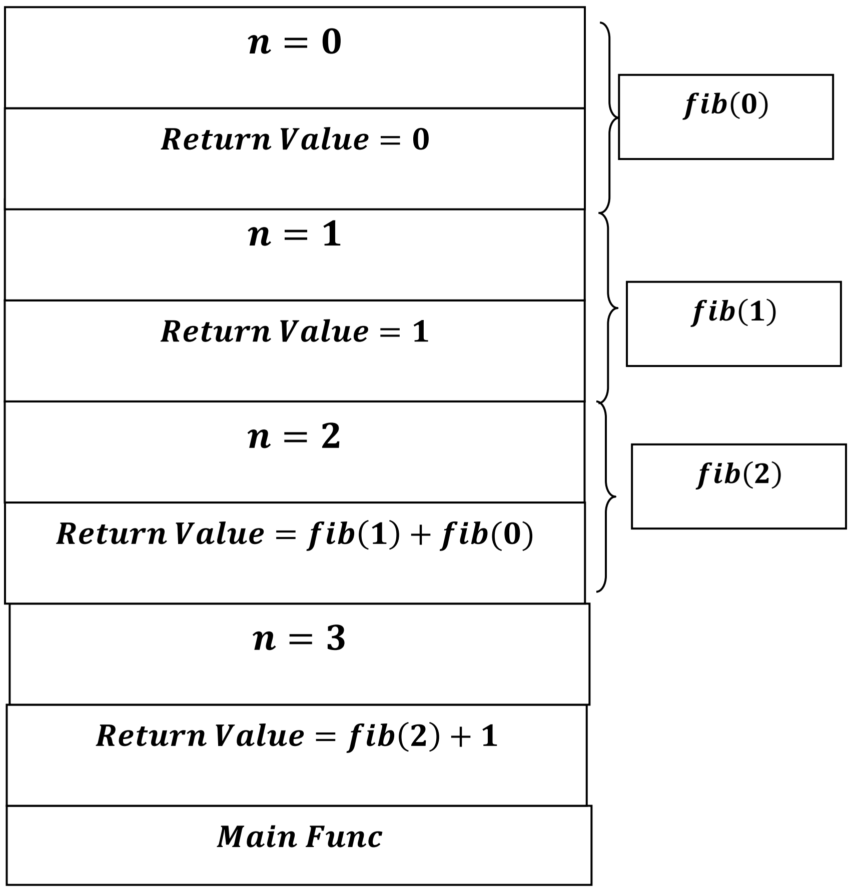


$\mathbf{4.1.}\,\boldsymbol{pop}$

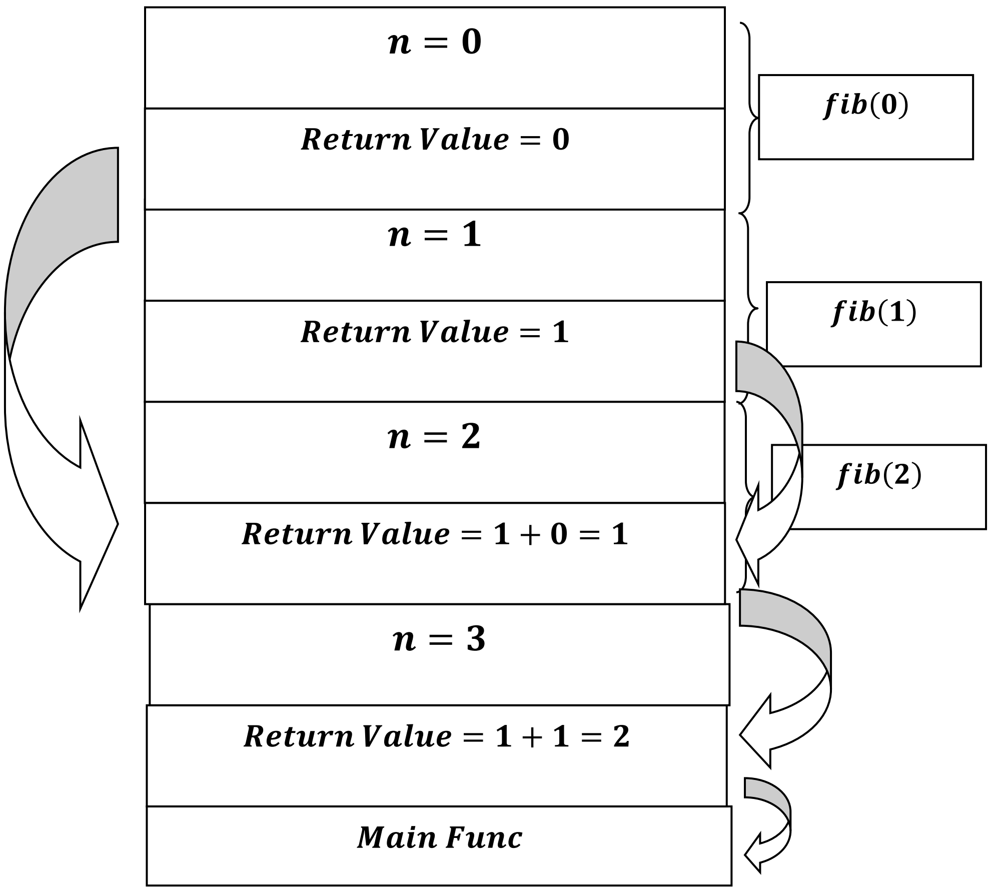


$\boldsymbol{Now\ fibonacci(2)\ will\ return\ again : fibonacci(1) + fibonacci(0)}$

$4.2.\ push$



$4.2.\ pop$



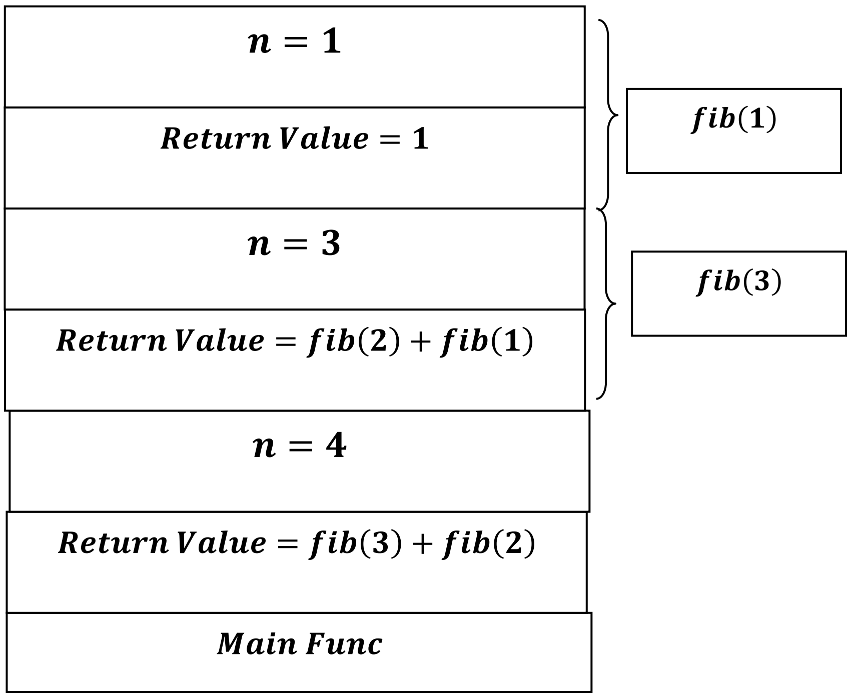
$$\boldsymbol{Hence\ upto\ now\ we\ got\ series: 0\ , 1, 1, 2}$$

$\boldsymbol{fibonacci(4)\ , we\ get:}$

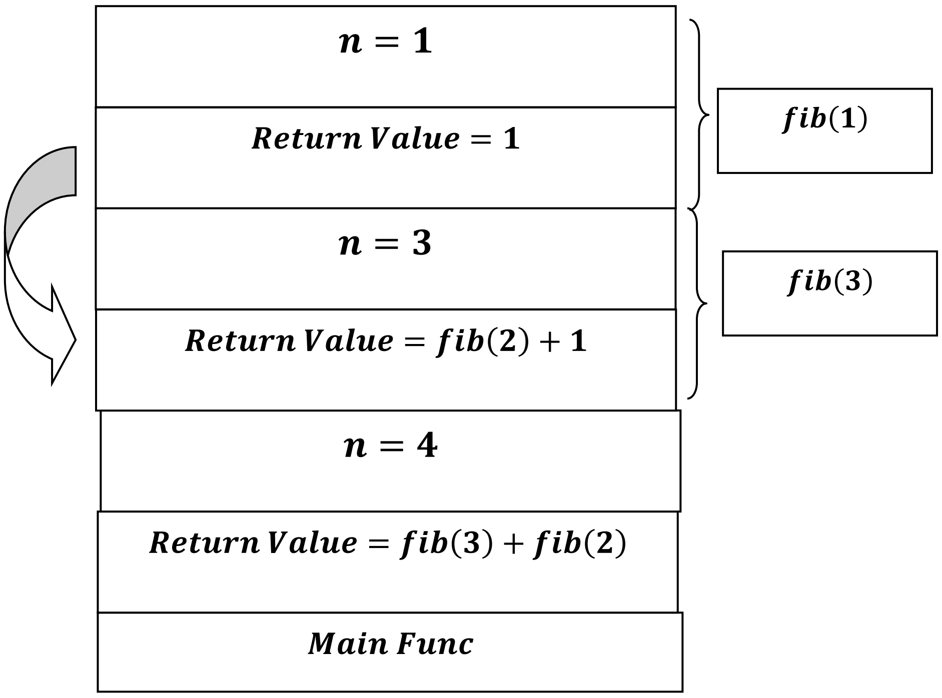
$\boldsymbol{fibonacci(4)\ will\ return:\ fibonacci(3) + \ fibonacci(2)}$

$\boldsymbol{fibonacci(3) will\ return: fibonacci(2) + fibonacci(1)}$

$\mathbf{5.1.}\boldsymbol{push}$

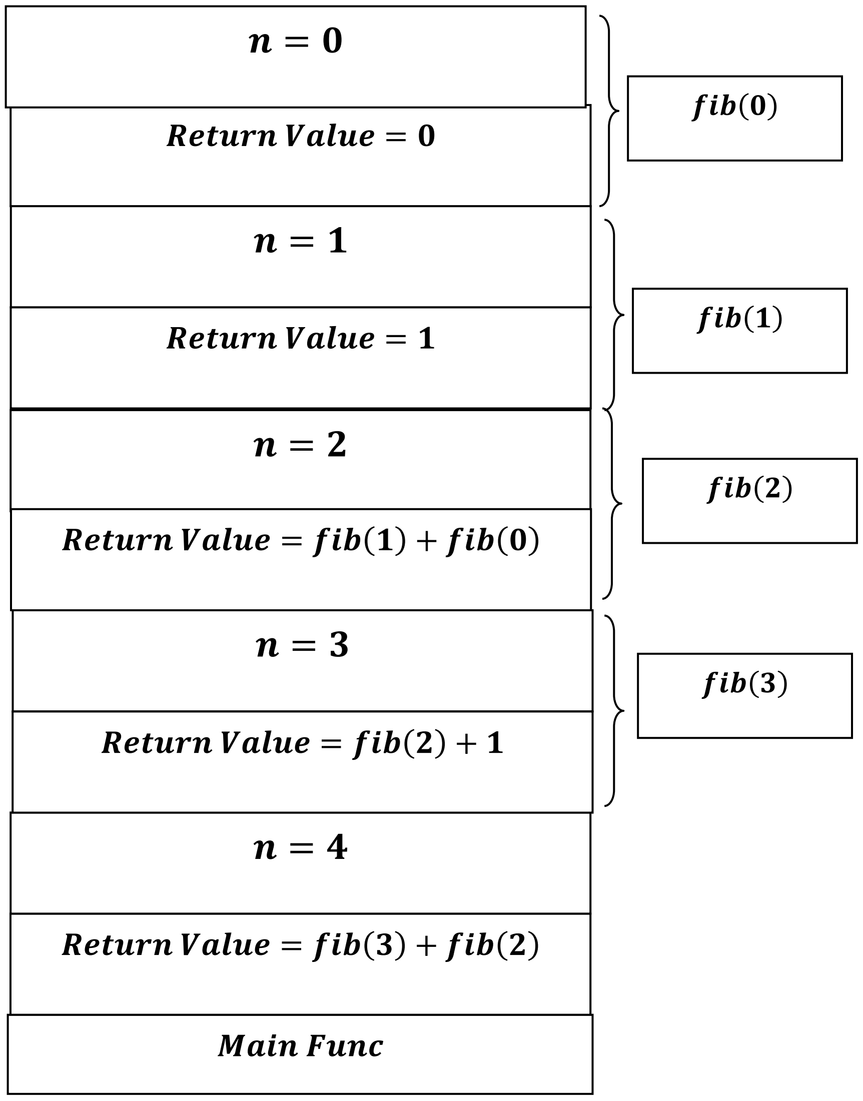


$\mathbf{5.1.}\boldsymbol{pop}$

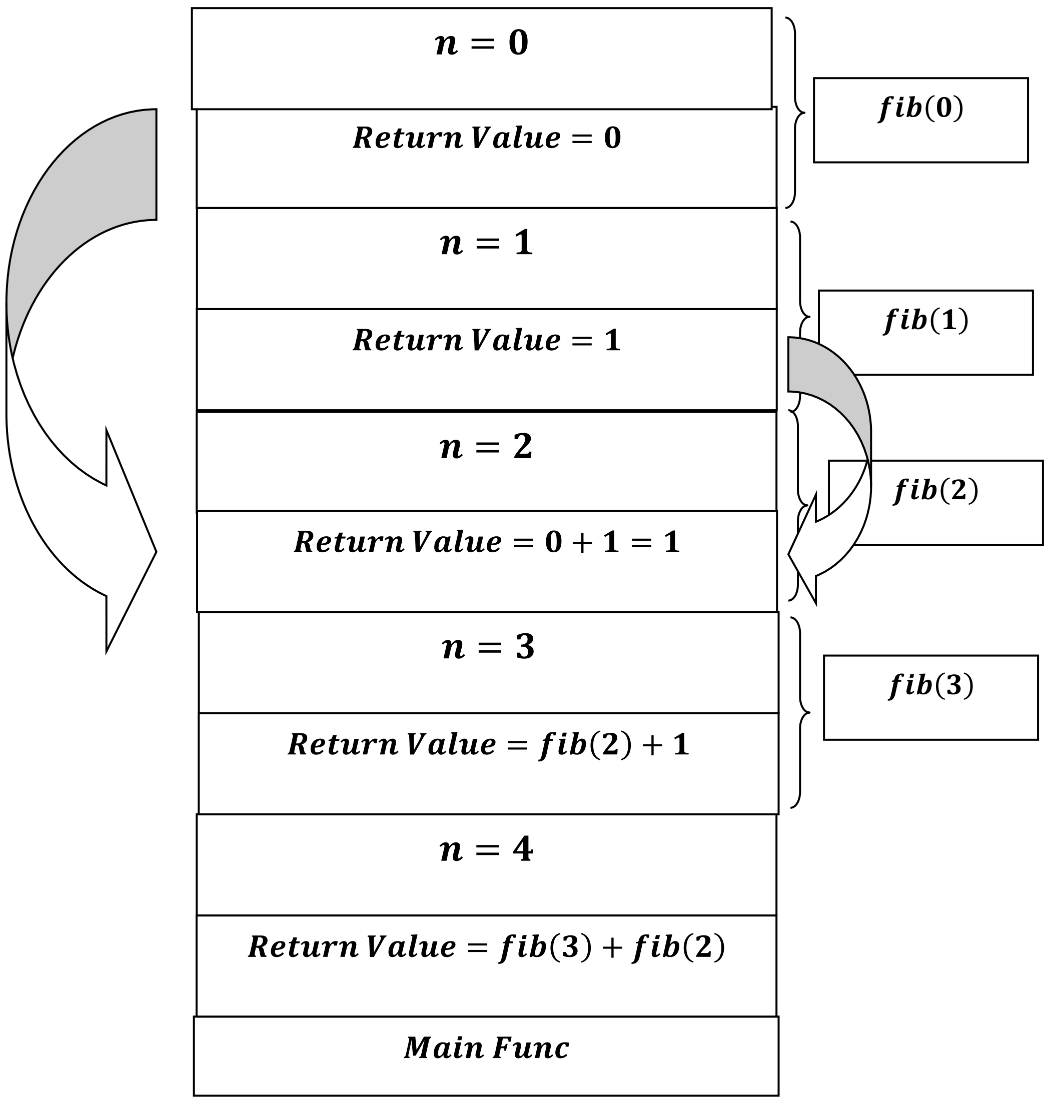


$$\boldsymbol{fibonacci(2)\ will\ return = fibonacci(1) + fibonacci(0)}$$

$\mathbf{5.2.}\boldsymbol{push}$

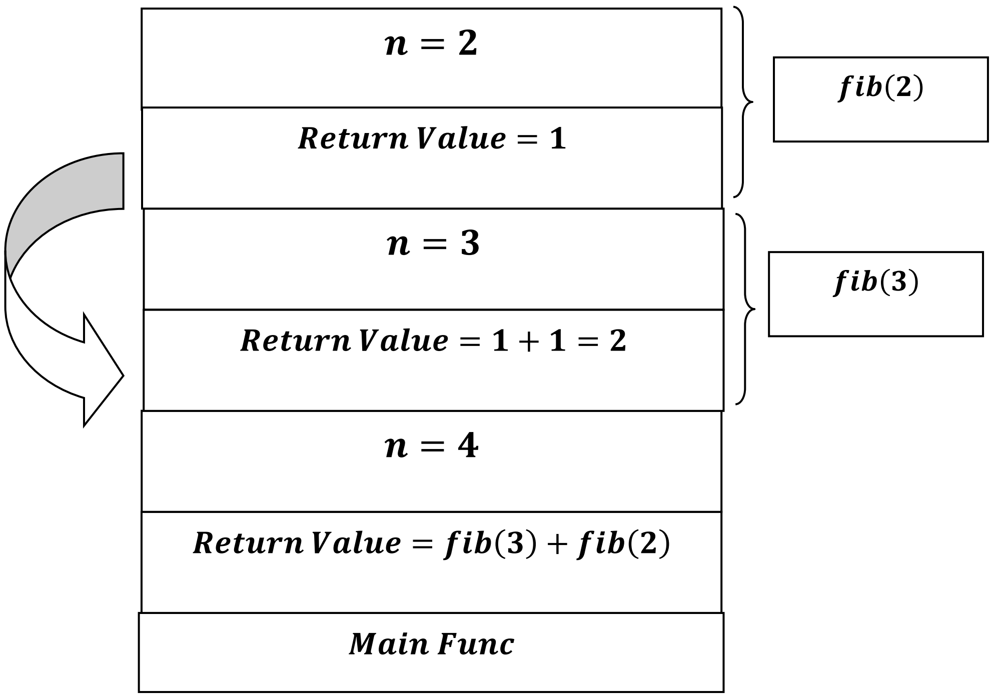


$\mathbf{5.2.}\boldsymbol{pop}(\mathbf{1})$

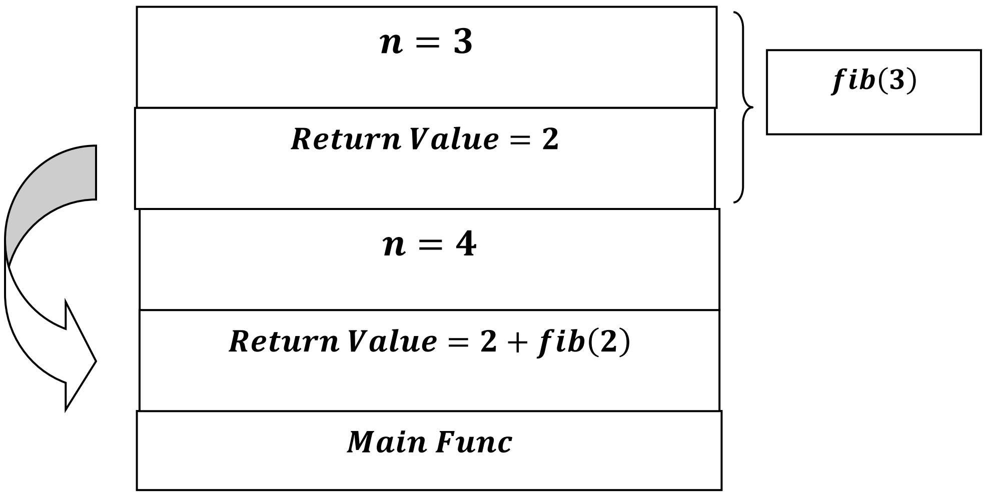


$n = 0$
$Return\ Value = 0$
$n = 1$
$Return\ Value = 1$
$n = 2$
$Return\ Value = 0 + 1 = 1$
$n = 3$
$Return\ Value = fib(2) + 1$
$n = 4$
$Return\ Value = fib(3) + fib(2)$
$Main\ Func$
$fib(0)$
$fib(1)$
$fib(2)$
$fib(3)$

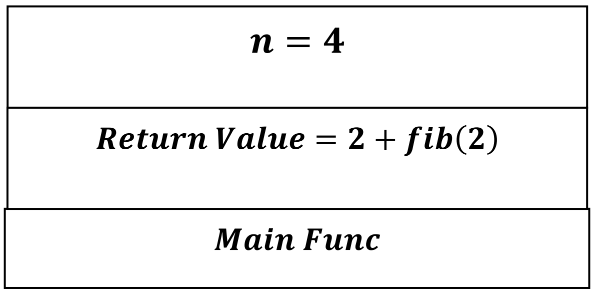
$5.2. pop(2)$



$5.2. pop(3)$

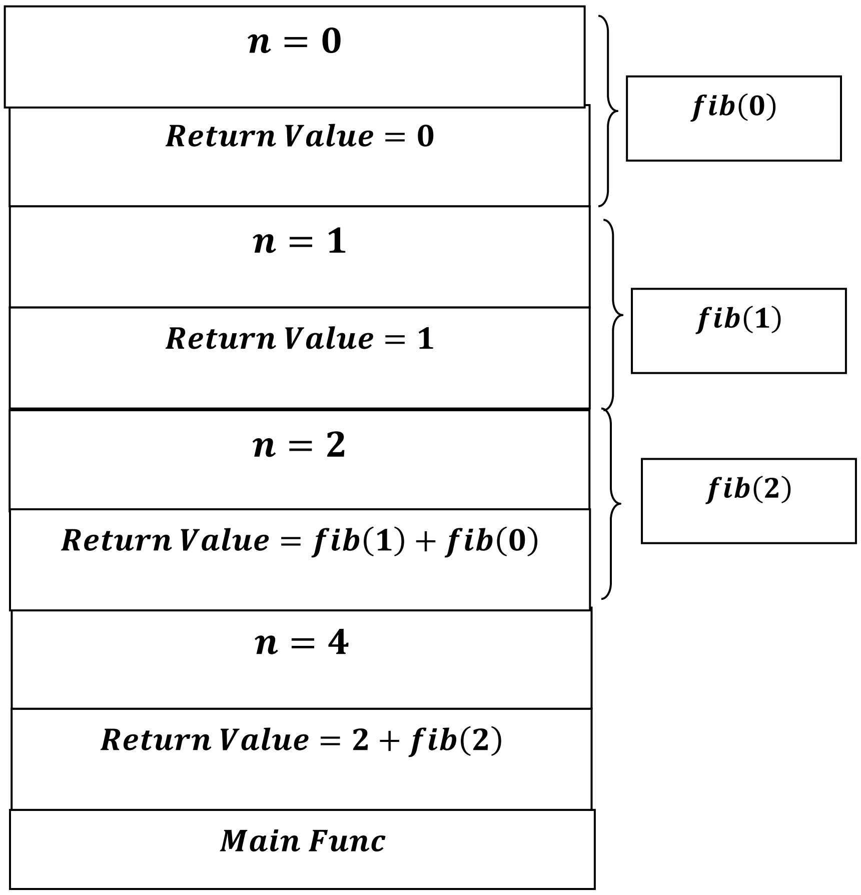


Hence at present we have stack after pop function :

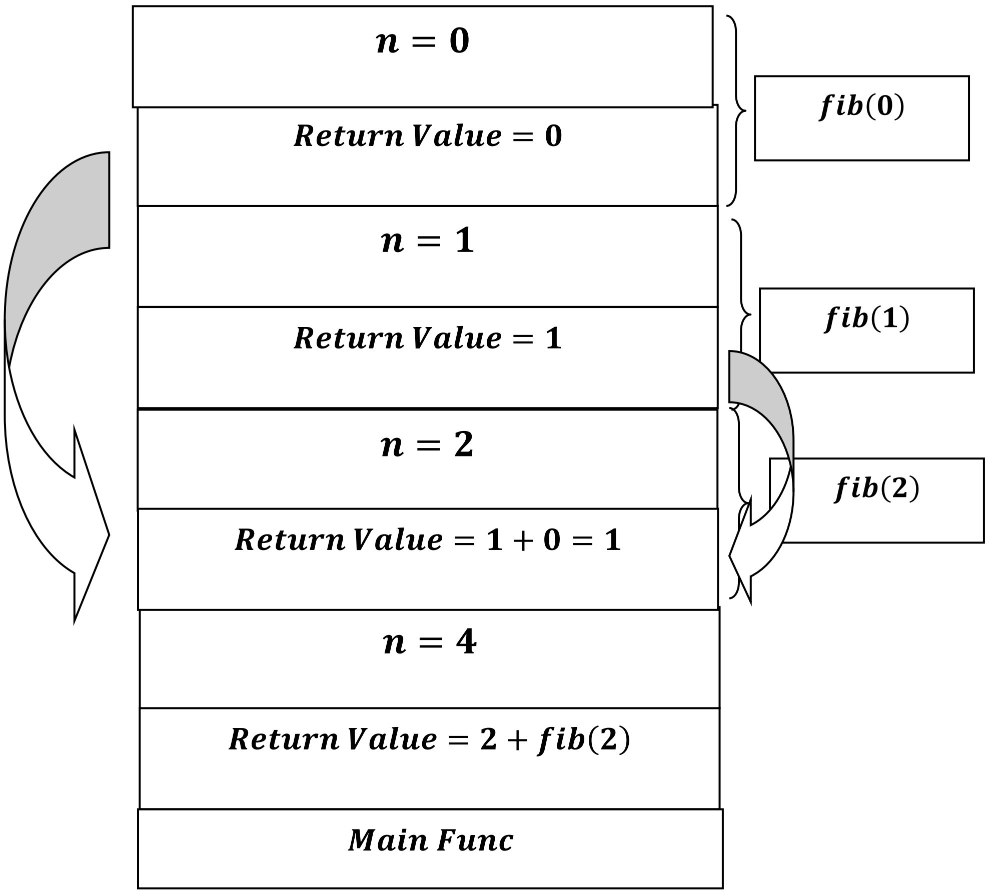


$$Again\ ,fibonacci(2) = fibonacci(1) + fibonacci(0)$$

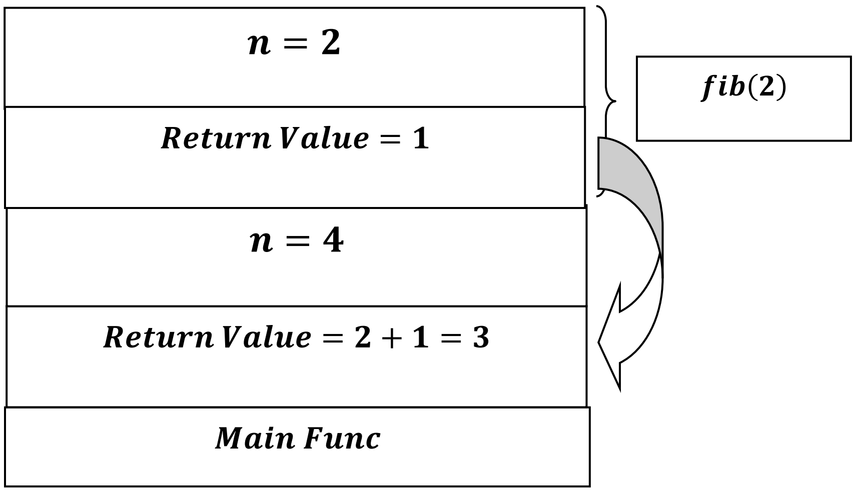
$6.1\ push$



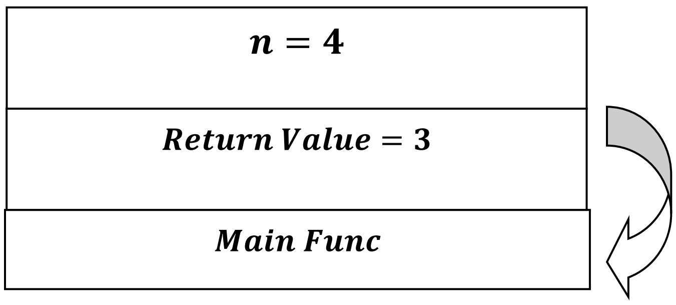
$6.2\ pop(1)$


$n = 0$
$Return\ Value = 0$
$fib(0)$
$n = 1$
$Return\ Value = 1$
$fib(1)$
$n = 2$
$Return\ Value = 1 + 0 = 1$
$fib(2)$
$n = 4$
$Return\ Value = 2 + fib(2)$
$Main\ Func$

$6.2\ pop(2)$


$n = 2$
$Return\ Value = 1$
$fib(2)$
$n = 4$
$Return\ Value = 2 + 1 = 3$
$Main\ Func$

$6.2\ pop(3)$

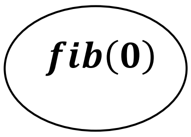


$Hence\ , now\ we\ get\ sequence: 0, 1, 1, 2, 3$

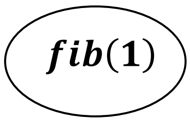
$Hence\ first\ five\ (5)\ sequence\ we\ get: 0, 1, 1, 2, 3$

$Hence\ recursive\ tree\ approached\ from\ here\ are:$

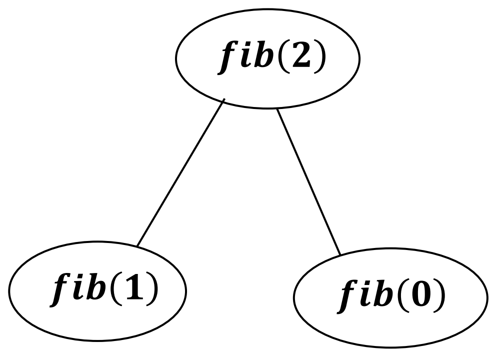
$1)\ When\ fibonacci(0)\ , we\ get\ a\ single\ node\ i.e.:$



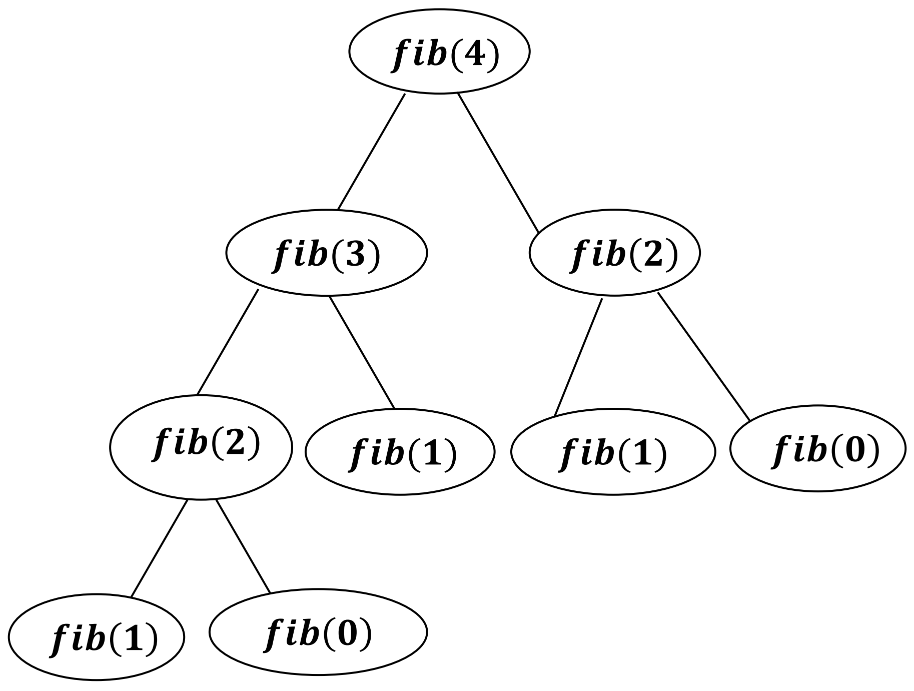
$2)\ When\ fibonacci(1)\ , we\ get\ a\ single\ node\ i.e.:$



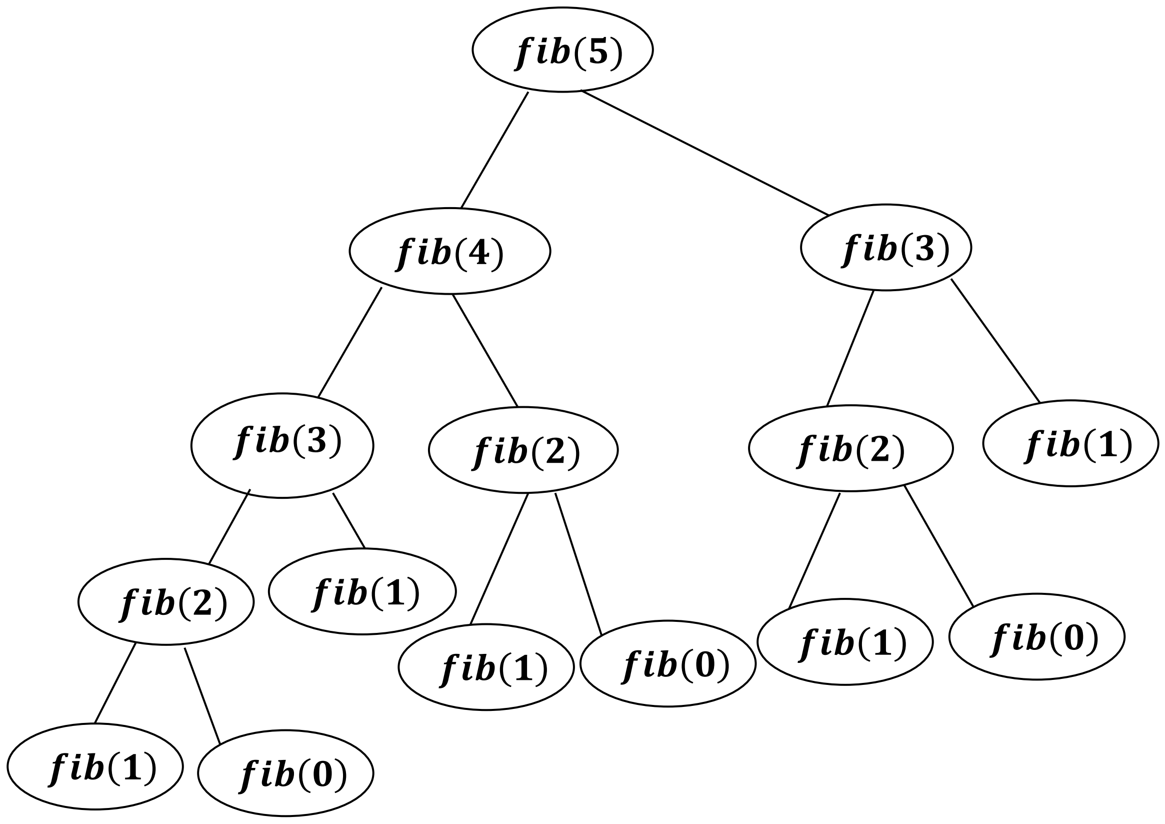
3) $\boldsymbol{When\ fibonacci(2)\ ,we\ get\ recursion\ tree}$:



4) $\boldsymbol{When\ fibonacci(4)\ ,we\ get\ recursion\ tree}$:



$\boldsymbol{In\ Addition: fibonacci(5)\ , we\ get\ recursion\ tree} \rightarrow$



****