

DATA STRUCTURES & ALGORITHMS

24: DYNAMIC PROGRAMMING

(FIBONACCI SERIES)

Dr Ram Prasad Krishnamoorthy

Associate Professor
School of Computing and Data Science

ram.krish@saiuniversity.edu.in



Generating first ***n*** terms
of Fibonacci Sequences

RECURRENCE RELATION

Fibonacci numbers

In mathematics, Fibonacci number denoted F_n form a sequence called Fibonacci sequence, such that each number is the **sum of preceding two numbers**, starting from 0 and 1.

$$F_0 = 0, \quad F_1 = 1,$$

and

$$F_n = F_{n-1} + F_{n-2}$$

for $n > 1$.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...



Fibonacci is a made up name for Italian mathematician *Leonardo Bonacci of Pisa*. The name Fibonacci comes from the Latin "*filius **Bonacci***", "the son of Bonacci". His father was Guglielmo Bonaccio.

The sequence had been described earlier in Indian mathematics, as early as **200 BC** in work by **Pingala** on enumerating possible patterns of Sanskrit poetry formed from syllables of two lengths. ([Wikipedia](#))

RECURSIONS - Fibonacci Sequence

```
#include <stdio.h>

int fibonacci(int);

int main(void)
{
    int n = 10;

    for(int i = 0; i < n; i++)
    {
        printf("%d ", fibonacci(i));
    }
    printf("\n");

    return 0;
}
```

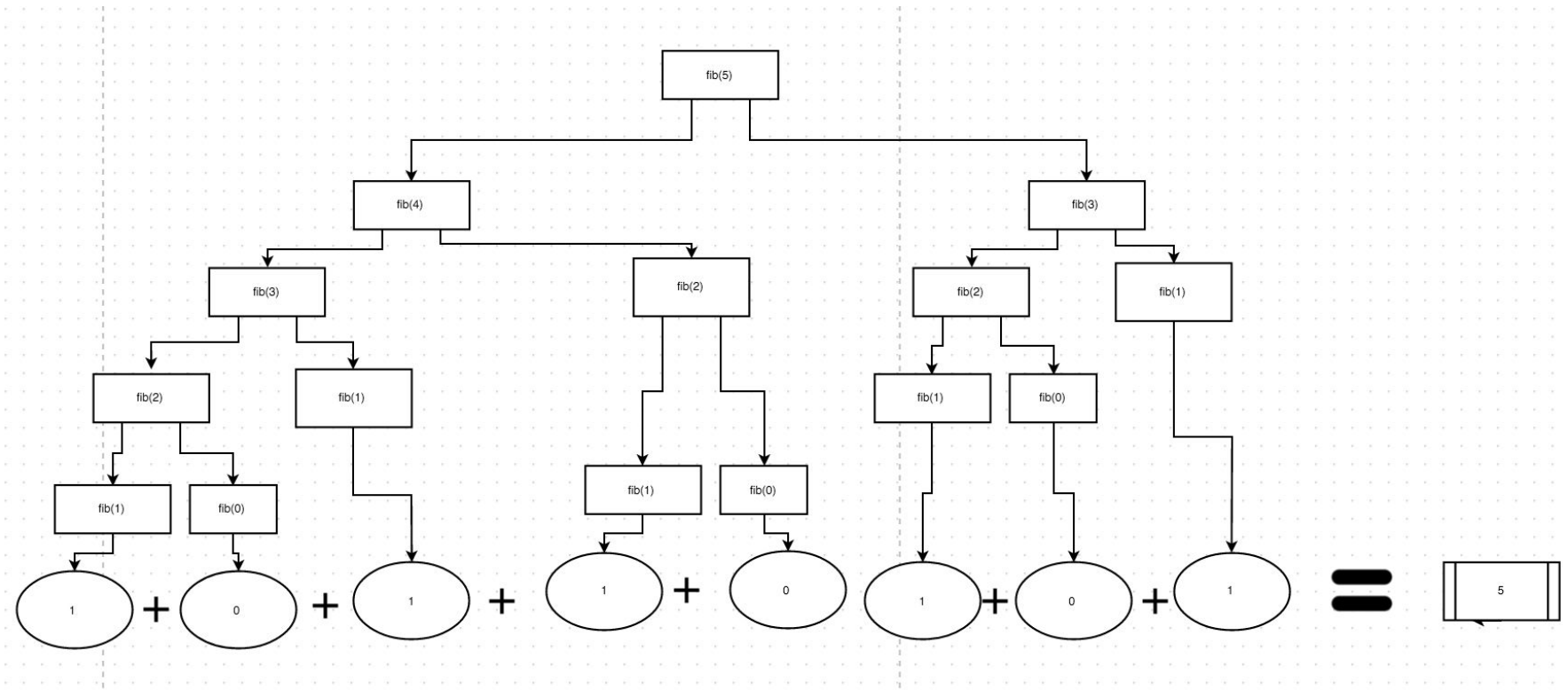
```
int fibonacci(int i)
{
    if(i == 0)
        return 0;

    else if(i == 1)
        return 1;

    else
        return (fibonacci(i-1) + fibonacci(i-2));
}
```

0 1 1 2 3 5 8 13 21 34

RECURSIONS - Fibonacci Sequence



<https://stackoverflow.com/a/34008122>

DYNAMIC PROGRAMMING

DYNAMIC PROGRAMMING

Dynamic Programming

- Method for solving complex problems by breaking them down into simpler sub-problems.
- It is applicable to problems exhibiting the properties of overlapping sub-problems and optimal substructure.
- A dynamic programming algorithm will examine all possible ways to solve the problem and will pick the best solution.
- Uses a technique called memoization which stores the intermediate results in a table.

DYNAMIC PROGRAMMING

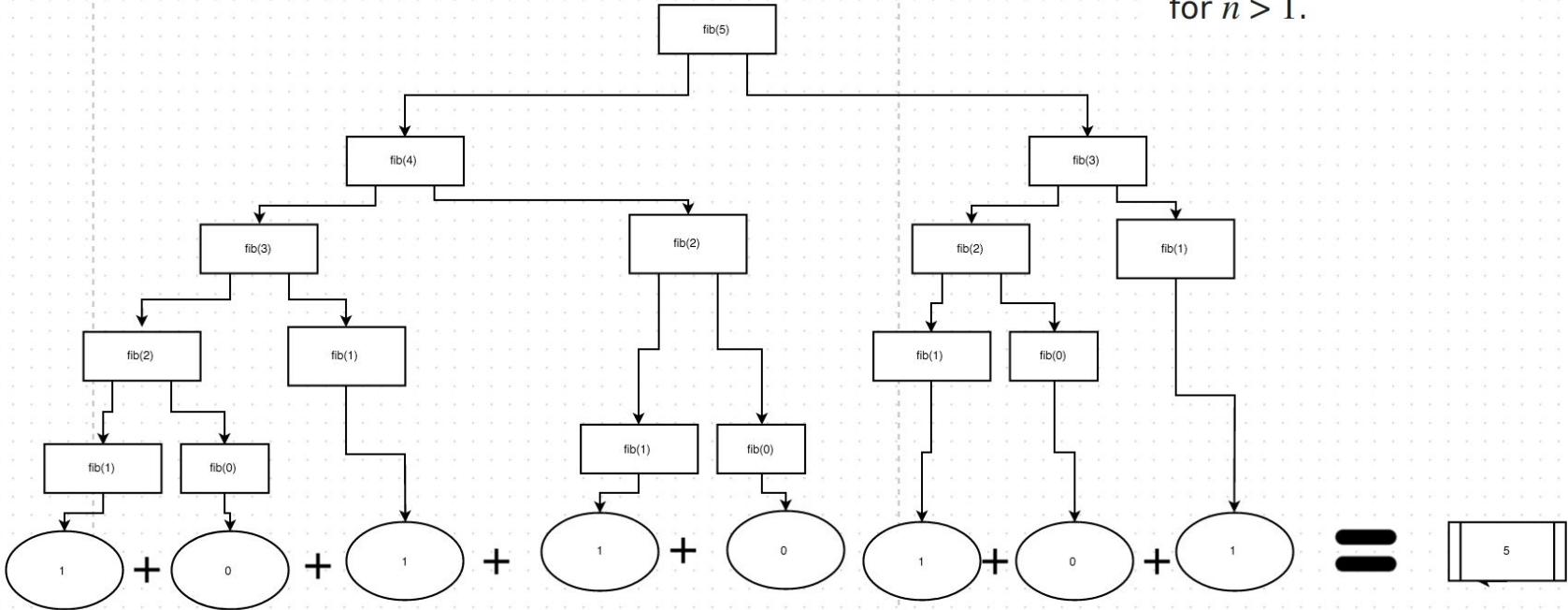
Fibonacci numbers

$$F_0 = 0, \quad F_1 = 1,$$

and

$$F_n = F_{n-1} + F_{n-2}$$

for $n > 1$.



DYNAMIC PROGRAMMING

```
# nth Fibonacci number using Dynamic Programming
```

```
fTable = [0, 1]    # Used for memoization
```

```
def fib(n):  
    if (n < 0):  
        print("Invalid n")  
    elif (n < len(fTable)):  
        return fTable[n]  
    else:  
        val = fib(n-1) + fib(n-2)  
        fTable.append(val)  
        return val
```

```
v = fib(10)  
print(v)  
print(fTable)
```

55
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55]