

Name : Mollah Md. Saif

Lab Assignment -01

Id no : 20101416

Course : CSE 221

Section : 10

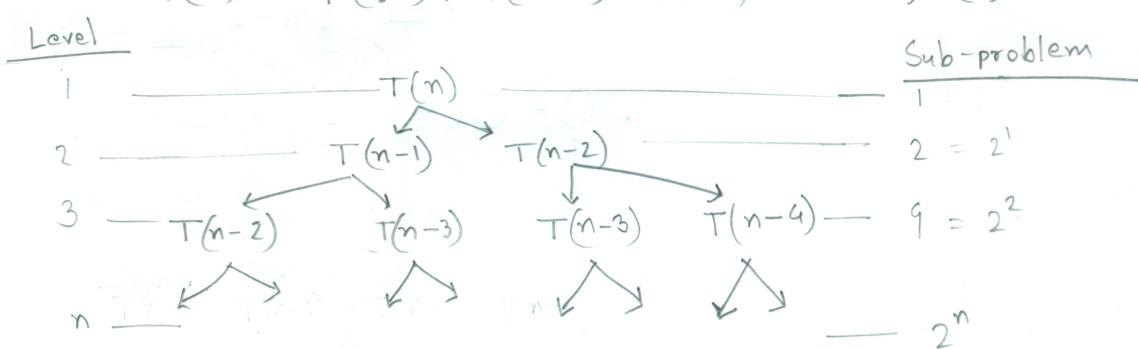
## Problem 2

### Implementation - 1

```
def fibonacci_1(n):
    if n <= 0:
        print("Invalid input!")
    elif n <= 2:
        return n-1
    else:
        return fibonacci_1(n-1) + fibonacci_1(n-2)
```

From the above code we get,

$$T(n) = T(n-1) + T(n-2) + C ; T(2)=1; T(1)=0$$



$\therefore$  There is  $2^n$  problems. The time complexity is  ~~$O(n^2)$~~   $O(2^n)$

## Implementation - 2

```
def fibonacci_2(n):
    Fibonacci_array = [0, 1] → 1
    if n > 0: → 1
        | print("Invalid input!")
    elif n <= 2: → 1
        | return Fibonacci_array[n-1]
    else: → 1
        | for i in range(2, n): → (n-2)
            | | Fibonacci_array.append(Fibonacci_array[i-1] + Fibonacci_array[i-2])
    return Fibonacci_array[-1] → 1
```

From the code we get,

$$\begin{aligned} \text{Time complexity, } T(n) &= (1+1+1+(n-2) \times 2 + 1) \\ &= 2n-2 \end{aligned}$$

$$T(n) = O(n)$$

## Problem 4

Pseudocode:

Procedure Multiply-matrix (A, B)

    Input A, B  $n \times n$  matrix

    Output C  $n \times n$  matrix

begin

    Initialize C as a  $n \times n$  zero matrix  $\longrightarrow 1$

    for i = 0 to  $n-1$   $\longrightarrow n$

        for j = 0 to  $n-1$   $\longrightarrow n$

            for k = 0 to  $n-1$   $\longrightarrow n$

$c[i, j] = A[i, k] * B[k, j]$   $\longrightarrow 1$

            end for

        end for

    end for

end Multiply-matrix

$$\therefore \text{The time complexity is } T(n) = 1 + n \times n \times n \times 1 \\ = n^3 + 1$$

$$\therefore T(n) = O(n^3)$$