1. sum(N): $\longrightarrow$ s(n)

```
if N==1:
    return 1
else:
    return N + sum(N-1)
```

if N==1 / return 1 $\longrightarrow$ 1

else / return N + sum(N-1) $\longrightarrow$ $s(n-1)+n$

$\downarrow$

$s(n-1)$

$$s(n) = \begin{cases} 1 ; & n=1 \\ s(n-1)+1 ; & n>1 \end{cases}$$

$$s(n) = s(n-1) + 1$$

$a=1, b=1$

Case 2: $a==1$: $O(n * f(n))$

$\Rightarrow O(n * 1) \Rightarrow O(n)$

2. Prod(n): $\longrightarrow$ P(n)

```
if n==1:
    return 1
else:
    return Prod(n-1)*n
```

if n==1 / return 1 $\longrightarrow$ 1

else / return Prod(n-1)*n $\longrightarrow$ P(n-1) +1

Page No. ______
Date ______

~~Prod~~

$$P(n) = \begin{cases} \cancel{1; n=1} \\ P(n-1) + 1; n>1 \end{cases}$$

$$P(n) = P(n-1) + 1$$

$a=1$

Case 2 : $a==1$: $O(n \times f(n)) \Rightarrow O(n)$

3. fibonacci(n) $\longrightarrow$ F(n)

```
if n==1 :
    return 0
elif n==2 :
    return 1
else :
    return fibonacci(n-1) fibonacci(n-2)
```

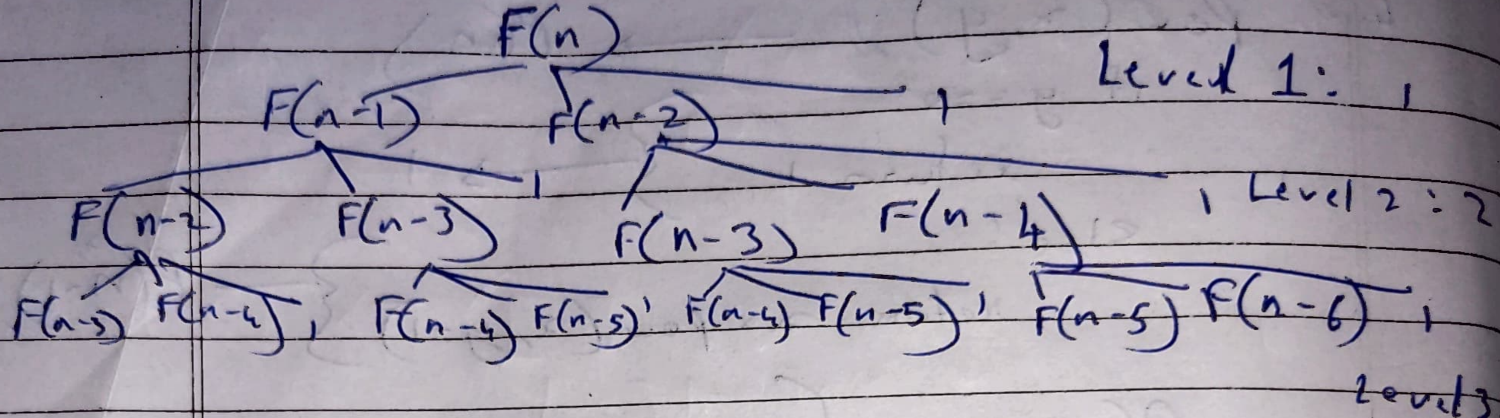
$\longrightarrow 1$

$$F(n) = \begin{cases} 1 ; & n<=2 \\ F(n-1) + F(n-2) + 1 ; & n>2 \end{cases}$$

F(n-1) + F(n-2)

$$F(n) = F(n-1) + F(n-2) + 1$$

Remarks ______



Level 4 : 8

Level k : $1 + 2 + 4 + 8 + \cdots \to 2^{k-1}$

$\Rightarrow a = 1$
$r = 2$

Sum of k terms in G.P $= \dfrac{a(r^k - 1)}{(r-1)}$

$\Rightarrow \dfrac{1(2^k - 1)}{2-1} = 2^k$

$F(n) = F(n-2k) + 2^k - 1$

$F(n-2k) = F(2) = 1$

$n - 2k = 2$

$n = 2k + 2$

$k = \dfrac{n-2}{2}$

$F(n) = \cancel{1} + 2^{\frac{n-2}{2}} \cancel{-1} = 2^{n/2 - 1}$

$= \dfrac{2^{n/2}}{2}$

Remarks

$\Rightarrow O(2^n)$

Page No. ______
Date ______

4. Pow(x, y): $\longrightarrow$ P(n)
    if y == 0:
        return 1 $\Big\} \longrightarrow$ 1
    else:
        return x × pow(x, y-1) $\Big\}$ — P(n-1)

$$P(x, n) = \begin{cases} 1 ; & y = 0 \\ P(n-1)+1 ; & y > 0 \end{cases}$$

$\underline{P(n) = P(n-1) + 1}$

Case 2: $a = 1$

$O(n \cdot f(n)) \Rightarrow O(n)$ //

5. & 6. natural(n): $\longrightarrow$ T(n)
    if n == 1:
        Print(1)
        return $\Big\} \longrightarrow$ 1
    else:
        natural(n-1)
        Print(n) $\Big\}$ — T(n-1)

$T(n) = T(n-1) + 1$

$\Rightarrow O(n)$

Remarks ______

Page No.
Date

7. GCD (a, b): $\rightarrow G(n)$

if b == 0 :

return a $\rightarrow 1$

else :

return GCD (b, a%b) $\rightarrow G(n/2)$

$$G(a) = \begin{cases} 1 & ; b == 0 \\ G(n/2) + 1 & ; b > 0 \end{cases}$$

$$G(n) = G(n/2) + 1$$

$a = 1 \quad k = 0$

$b = 2 \quad p = 0$

$\log_b a = \log_2 1 = 0$

$k = 0$

Case 2: $\log_b a == k$

if $p > -1 \Rightarrow O(n^k \log^{p+1} n)$

$\Rightarrow O(\log n)$

Remarks

Page No.
Date

9. ArrDisp(A) $\longrightarrow D(N)$

10. if A == [] : $\Big\} \rightarrow 1$

return

else:

ArrDisp(A[1:]) $\rightarrow D(N-1)$

print (A[0]

where $N = len(A)$

$$D(N) = \begin{cases} 1 ; & N = 0 \\ D(N-1)+1 ; & N > 1 \end{cases}$$

$$D(N) = D(N-1) + 1$$

$$\Rightarrow O(n)$$

10 revnum (N, r): $\longrightarrow R(K)$

if N == 0 : $\Big\} \rightarrow 1$

return r

else:

return revnum(N//10, r*10 + N%10) $\rightarrow R(K-1)$

where $K = \text{int}(\log_{10} N)$

Remarks

$$R(k) = \begin{cases} 1 ; & k = 1 \\ R(k-1)+1 ; & k > 1 \end{cases}$$

$$R(k) = R(k-1) + 1$$

$$\Rightarrow O(k) \Rightarrow O(\log N)$$

11. checkSorted (A) : $\longrightarrow C(N)$

```
checkSorted (A) :
    if len(A) < 2:
        return True
    elif A[0] > A[1]:
        return False
    else:
        return checkSorted(A[1:])
```

$\underbrace{\text{if} \ldots \text{return False}}_{1}$

$\text{return checkSorted}(A[1:]) \longrightarrow C(N-1)$

$$C(N) = \begin{cases} 1 ; & N < 2 \\ C(N-1)+1 ; & N \geq 2 \end{cases}$$

where N is the length of Array

$$C(N) = C(N-1) + 1$$

$$\Rightarrow O(n)$$

Remarks ______