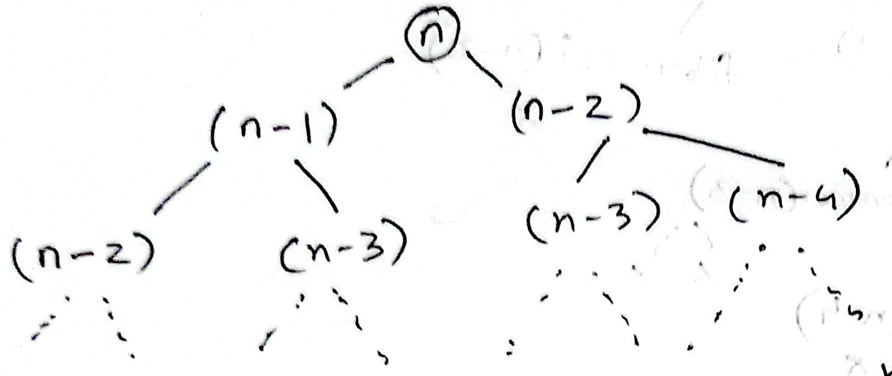
SADAF M. ANIS 20101537

221-LAB 1

Ans. to the question no. 02 :-

For the given implementation-1



$2^0 = 1$, step 0

$2^1 = 2$ step 1

$2^2 = 4$ step 2

(n-k) - - - - - - - - - - - - - - - - 2^k, step k. Here k = n.

For k steps it works 2^k on 2^n times.

$\therefore$ Total $= 2^0 + 2^1 + 2^2 + 2^3 + \cdots\cdots + 2^n$

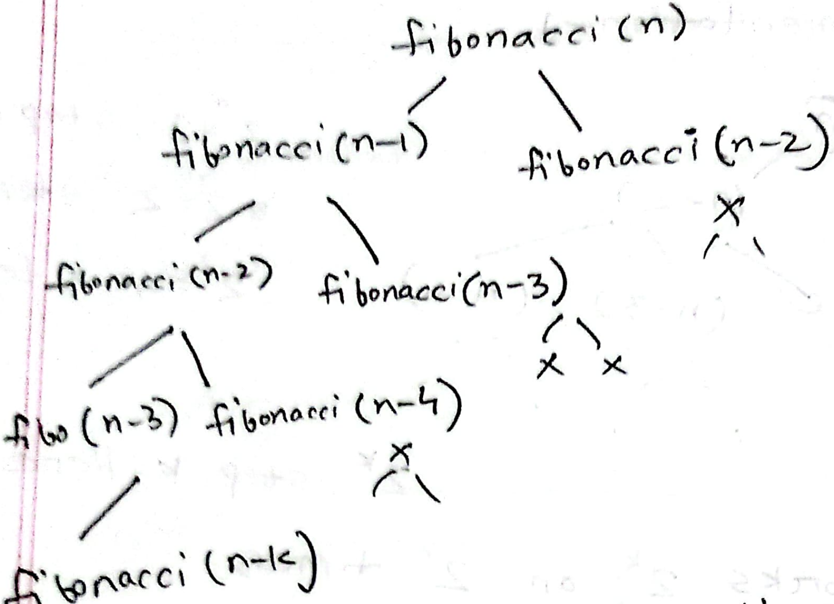
$= 2^{n+1} - 1$

$= 2^n$

$\therefore O(2^n) \rightarrow$ Time complexity of implementation 1

20101537

For implementation 2,



The algorithm is called 1 time for each value from 0 to n.

$\therefore$ For K steps the work is $= k * 1$
$= n * 1$

So, The Time complexity : $O(n)$

By comparing the both implementation, we see that 2nd implementation for each value for n, the function is making one branch. And in the first implementation there was two branch making by the function. Thus, the implementation 2 is faster and time convenient.

20101537

Ans. to the question no. 04:

for i=0 to n-1 # O(n)
for j=0 to n-1 # O(n)
for k=0 to n-1 # O(n)

[3 nested loops]

so, time complexity $= O(n^3)$ (Ans:)

Ans. to the question no. 05:

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

$$= T\left(\frac{n}{2^2}\right) + \frac{n}{2} + n - 1 - 1$$

$$= T\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2} + n - 1 - 1 - 1$$

$$= T\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}} + \frac{n}{2^{k-2}} + \dots \frac{n}{2} + n - k$$

Let, $\frac{n}{2^k} = 1$

$\therefore k = \log n$ & $n = 2^k$

$$\text{so, } T(n) = T(1) + n\left[\frac{1}{2^{k-1}} + \frac{1}{2^{k-2}} + \dots + 1\right] - \log$$

$$= T(1) + n\left[\frac{1}{2^{\log n - 1}} + \frac{1}{2^{\log n - 2}} + \dots + 1\right] - \log n$$

$$= T(1) + n[1+1] - n\log n \left\{\frac{1}{2^{\log n - 1}} + \frac{1}{2^{\log n - 2}} + \dots + 1\right\}$$

$\therefore T(n) = O(n)$ (Ans:)

2010537

or using master theorem,

$$T(n) = T(\frac{n}{2}) + n - 1$$

Here, $a = 1,\ b = 2,\ d = 1$

So, $1 < 2^1 \Rightarrow a < b^d$

$\therefore$ time complexity $O(n)$

② $T(n) = T(n-1) + n - 1$; $T(1) = 0$

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

$= T(n-3) + (n-2) + (n-1) + n - 3$

$\vdots$

$= T(n-n) + ((n-n+1) + 2 + 3 + 4 + \cdots n) - n$

$= T(0) + (1 + 2 + 3 \cdots + n) - n$

$= \frac{n(n-1)}{2} - n$

$= \frac{n^2 - n}{2} - n$

$= n^2$

$\therefore O(n^2)$

20101537

(3) $T(n) = T(n/3) + 2T(n/3) + n$

$= 3T(\frac{n}{3}) + n$

Using master theorem,

$a=3, b=3, d=1$

Here, $3 = 3^1$

$a = b^d$

So, time complexity $O(n \log n)$ Ans:)

(4) $T(n) = 2T(\frac{n}{2}) + n^2$

using master theorem,

$a=2, b=2, d=2$

Here, $2 < 2^2$

$\Rightarrow a < b^d$

So, time complexity $T(n) = O(n^2)$

[Proved]