of Algorithms

Tutorial- 2

Q1 What is the time Complexity of below Code and how?

```
void fun(int n)
{ int j=1, i=0;
  while (i<n)
    i=j+j;
    j++; }
```

→ $O(n)$

In this program it seems to be the time Complexity is $O(n)$ because the while loop will takes place upto n times.

But in actual.

j	1	2	3	4	...	n
i	1	3	6	10	...	k

$$\frac{k(k+1)}{2} > n$$

$$\frac{k^2+k}{2} > n$$

$$k^2 = n$$

$k = \sqrt{n}$

$T.C = O(k) = O(\sqrt{n})$

$$\boxed{T.C = O(\sqrt{n})}$$

<u>Solution 2)</u>

```
int fib ( int n)          → T(n)
{
    if (n<=1)             → O(1)
        return n;

    return fib(n-1) + fib (n-2);
                          → T (n-1) + T (n-2)
}
```

$$T(n) = \begin{cases} 1 & , \; n <= 1 \\ T(n-1) + T(n-2) & , \text{ otherw} \end{cases}$$

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

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

[from the approximation $T(n-1) \sim T$

$T(n) = 2(2T(n-2) + c) + c$

$T(n) = 4T(n-2) + 3c$

$= 8T(n-3) + 7c$

$\vdots$

$2^k T(n-k) + (2^k - 1)c$

We know that $T(1) = 1$, so

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

$n-k = 1$

$\boxed{k = n-1}$

We get

$T(n) = 2^{n-1} T(1) + (2^{n-1} - 1) c$

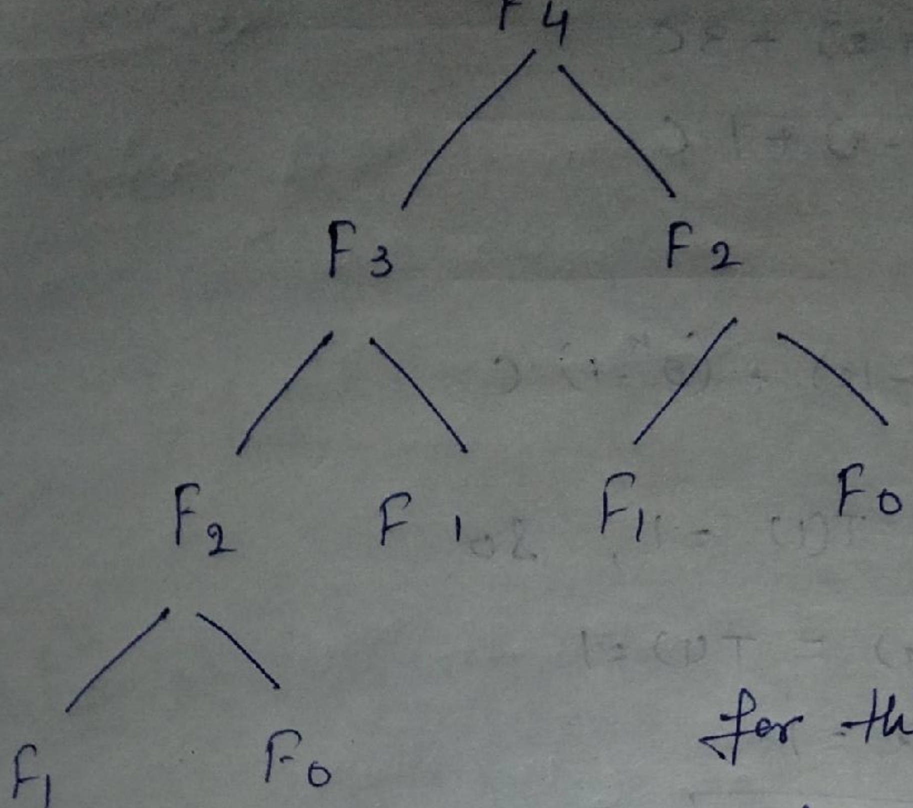
$\boxed{T.C = O(2^n)}$

Now, for space complexity we know that

$\boxed{\text{Space reqd} \propto \text{Max. depth of recursion Tree}}$

because that is the max. no. of elements that can be present in the implicit fxn call stack.

Now, lets take an Example F_4, So.



for this Space
comp = O(4)

for n-Elements { $\boxed{S.C = O(n)}$

Solution 3 > (i) T C = nlogn

```
A()
{ int i,j;
  for (i=1 ; i<=n ; i++)              → O (n)
    for (j=1 ; j<=n ; j=j*2)          → O(lo
    {
      printf ("Hey")
    }
  }
}
```

solution 4

$$T(n) = 2T(n/2) - cn^2$$

using master's method

$$Tn = aT(n/b) + f(n)$$

we get

$$C = \log_2^2 = 1$$

$$f(n) > n^c$$

$$T(n) = \Theta(f(n))$$

$$= O(n^2)$$

solution 5)

for $i = 1 \rightarrow j = 1, 2, 3, 4 \ldots n$ (sum fo n times)

for $i = 2 \rightarrow j = 1, 3, 5 \ldots$ sum for $n/2$ time)

for $i = 3 \rightarrow j = 1, 4, 7 \ldots$ (sum of $n/3$ times)

$$T(n) = n + n/2 + n/3 + n/4 + \ldots$$

$$n(1 + 1/2 + 1/3 + 1/4 + \ldots)$$

$$n \int_1^n \frac{1}{x} \rightarrow n \int \frac{dx}{x} \Rightarrow \log x \Big]^n$$

$$\boxed{TC = n \log n.}$$ Ans

Solution 6)

for first iteration $i = 2$

2^{nd} iteration $i = 2^k$

3^{rd} iteration $i = (2^k)^k = 2^{k^2}$

⋮

n^{th} iteration $i = 2^{k^i}$ loop

ends at $2^{k^i} = n$

apply log

$\log n = \log 2^{k^i}$

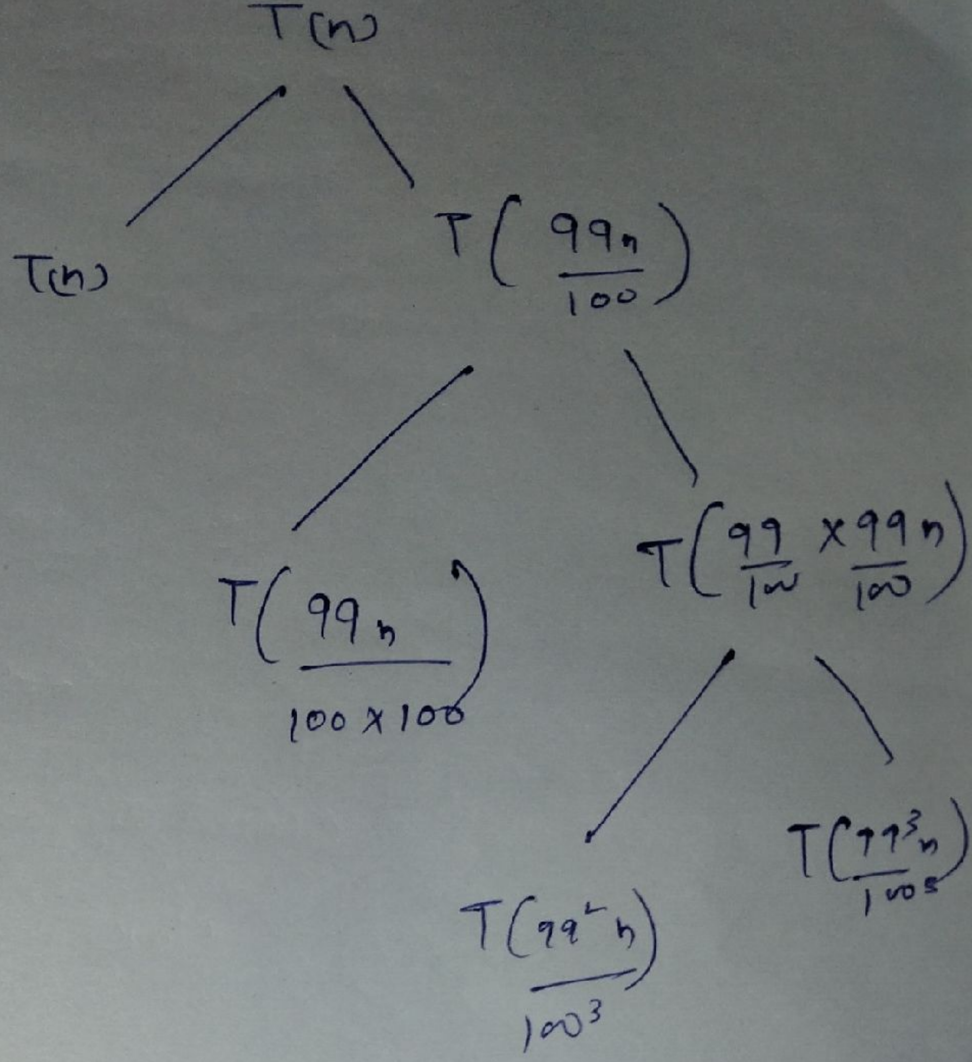
$k^i = \log n$

again log

$\log(k^i) = \log n$

$\boxed{T\ i = \log(\log n)}$

solutions :-



height of the tree = logn
no. of elements = n

$\boxed{T.C = n\log n}$

Solution 8 >

(a) $100 < \log(\log n) < \log(n) < \log^2 n < root(n)$
$< n < n\log n < n^2 < 2^n < 4^n < 2^{2n} < \log(n!)$
$< n!$

(b) $1 < \log(\log(n)) < \sqrt{\log n} < \log n <$
$\log 2n < 2\log n < n < 2n < 4n$
$< n\log n < n^2 < \log(n!) < n! < 2$

(c) $96 < \log_8(n) < \log_2(n) < 5n < n\log_6 n$
$< n\log_2 n < n! < \log n! < 8^{2n}$