Ans:-

$i = 0, 1, 3, 6, 10, 15, 21 \ldots n$

let the sum of above k term is S_K.

$S_K = 1 + 3 + 6 + 10 + \ldots + T_K$ — ①

$S_{K-1} = 1 + 3 + 6 + 10 + \ldots + T_{K-1}$ — ②

Subtract 2 from ①

$T_K = S_K - S_{K-1} = 1 + 2 + 3 + 4 + \ldots + K$

we have $T_K = n$

$\therefore 1 + 2 + 3 + 4 + \ldots + K = n$

$$\frac{K(K+1)}{2} = n \Rightarrow K^2 + K - 2n = 0$$

$$\Rightarrow K = \frac{-1 \pm \sqrt{8n+1}}{2}$$

taking only +ve value we get total no of times the loop runs for $i = K+1 = \frac{\sqrt{8n-1}}{2}$

$$TC = O\left(\frac{\sqrt{2n+1}}{2}^2\right) = O(\sqrt{n})$$

Teacher's Signature.....

Expt. No.....

Page No.....

Tutorial - 2

Design and Analysis of Algorithms

Ans 1) Let $T(0) = 1$

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

```
fib(n):
    if n <= 1
        return 1
    return fib(n-1) + fib(n-2)
```

Time Complexity :-

$$T(n) = T(n-1) + T(n-2) + c$$
$$= 2T(n-2) + c \quad \{ \text{let } T(n-1) \simeq T(n-2) \}$$

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

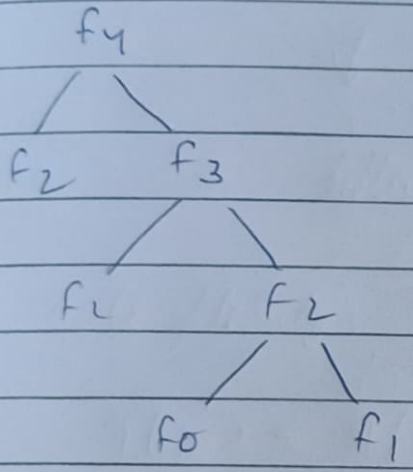
$$T(n-4) = 2^*(4T(n-4-4) + 3c) + c$$
$$= 8T(n-3) + 7c$$
$$= 2^k T(n-2k) + (2^k - 1)c$$

$n - 2k = 0 \Rightarrow n = 2k$, $k = n$

Teacher's Signature.....

$$T(n) = 2^n * T(0) + (2^n - 1)c$$
$$2^n * 1 + 2^n c - c$$
$$2^n(1+c) - c$$
$$\simeq 2^n \quad \text{// Constants ignored}$$
$$= O(2^n)$$

Space Complexity :- The space is proportional to the maximum depth of the recursion tree



Hence the space complexity of Fibonacci recursive is $O(n)$.

Ans 3:-

⇒ for time complexity $- n^3$

We can use three nested loops $- O(n^3)$

```
for(int i=0; i<n; i++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            Some O(1) expression.
```

Teacher's Signature.....

=> for time complexity - $\log(\log n)$

```
for(int i=2; i<n; i = pow(i,c))
    Some O(1) expression
```

=> for time complexity - $n \log n$

```
int fun(int n){
    for(i=1; i<n; i++)
        for(j=1; j<=n; j+=2)
            Some O(1) expression
```

Ans 4:-

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

Using master's method $T(n) = aT(n/b) + f(n)$
$a \geq 1, b > 1$ & $c = \log_b a$ comparing n^c & $f(n)$

We get $c = \log_2 2 = 1$

$f(n) > n^c$

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

$= O(n^2)$ //

Teacher's Signature.....

Expt. No.....

Page No.....

Ans 5->

for $i = 1 \rightarrow j = 1, 2, 3, 4 \ldots n$ (run for n times)
for $i = 2 \rightarrow j = 1, 3, 5 \ldots$ (run for $n/1$ times)
for $i = 3 \Rightarrow j = 1, 4, 7 \ldots$ (run for $n/3$ times)

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

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

$$= n \int_1^n \frac{1}{n} \Rightarrow n \int_1^n \frac{dx}{n} = n[\log n]_1^n$$

$$= n \log n$$

n.P

Ans b:- for first iteration $i = 2$
2nd iteration $i = 2^{\wedge k}$
3rd " $i = (2^k)^k = 2^{k^2}$
.
.
n^{th} " $i = 2^{k'}$ loop ends at $2^{k'} = n$

apply log $\log n = \log 2^{k'} \Rightarrow k' = \log n$
again apply log $\log(k') = \log n \Rightarrow i = \log_2(\log n)$.

Teacher's Signature.....

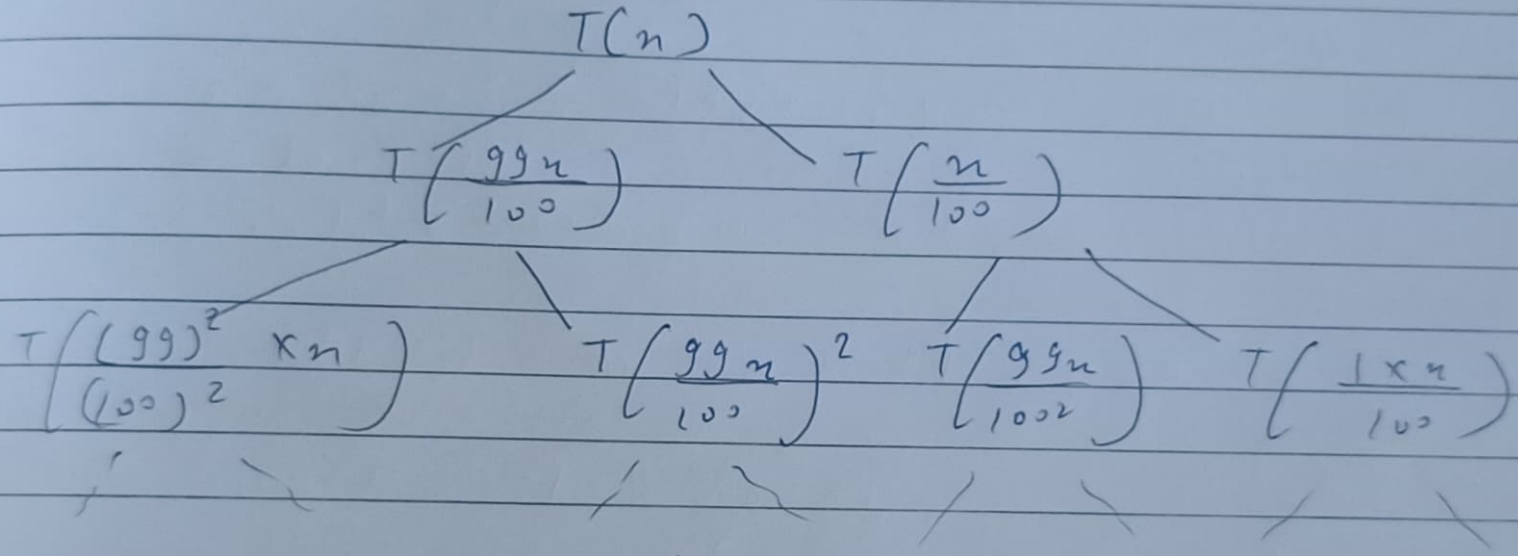
Sol 7:

99 61 in quick sort
where first is where from first or end always

So,

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(n/100\right) + O(n)$$

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(n/100\right) + O(n)$$



$$n = (99/100)^k$$

$$\log n = k \log 99/100$$

$$k = \log n \frac{100}{99}$$

$$TC = n^k \log_{\frac{100}{99}}(n)$$

Teacher's Signature.....

Date.....

Expt. No.....

Page No.....

Ans 8-a → $100 < \log\log(n) < \log^2 n < \log n < \log n! < n < n\log n$
$< n^2 < 2^n < 4^n < 2^{(2^n)} < n!$

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

Teacher's Signature.....