TUTORIAL-2

M T W T F S S
Page No.:
Date:
YOUVA

Q1. What is time complexity of :-

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

Series = 0, 1, 3, 6, 10, 15, ...

$n = 0 + 1 + 2 + 3 + \cdots + k$

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

$n \cong k^2$

$k \cong \sqrt{n}$

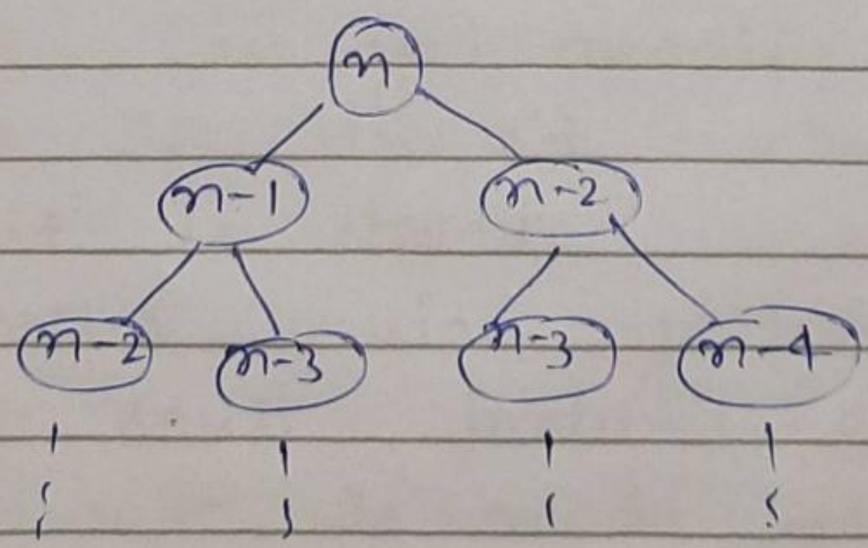
TC ⇒ $O(\sqrt{n})$

Q2:- Write recurrence relation for recursive function that prints fibonacci series. Solve recurrence relation to get time complexity of program. What will be the space complexity of this program and why?

Ans:- Recurrence Relation.

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

Using Tree Method:



—— 2

—— 2^2

..... 2^n

⇒ $1 + 2 + 4 + \cdots + 2^n = 1\left(\frac{2^{n+1}-1}{2-1}\right)$.

⇒ $2^{n+1} - 1$

TC = $O(2^n)$

Space complexity :- of fibonacci series using recursion is proportional to height of recurrence tree.

$SC = O(n)$

Q3. Write programs which have complexity.
$n(\log n)$, $n \wedge 3$, $\log(\log n)$

(i) $n(\log n)$

```
for (i to n)
{    for (j=1 ; j<=n ; j*=2)
                O(1)  statements
}
```

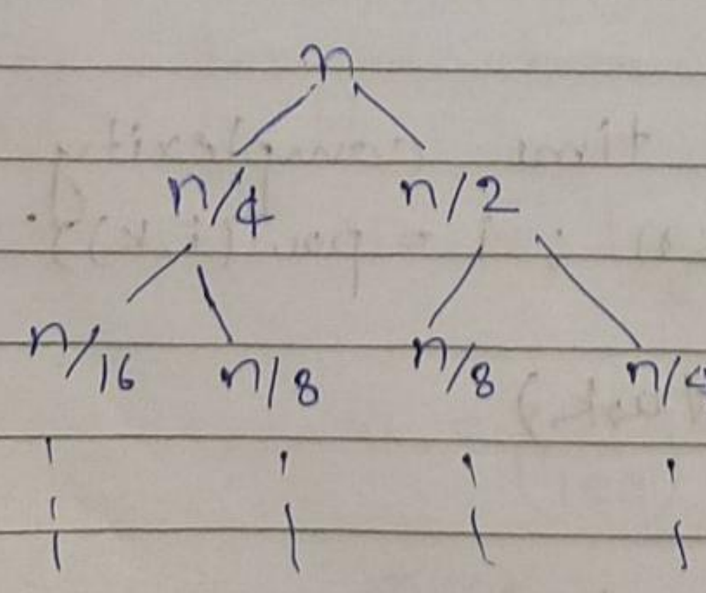
(ii) $n \wedge 3$

```
for (i to n)
        for (j to n)
                for (k to n)
                        O(1)  statements.
```

(iii) $\log(\log n)$

```
int  i=n;
while (i>0)
{        i=√i ;
}
```

Q(4) Solve $T(n) = T(n/4) + T(n/2) + cn\wedge 2$



$= n^2$

$= \frac{cn^2}{16} + \frac{cn^2}{4} = \frac{5cn^2}{16}$

$= \frac{cn^2}{256} + \frac{cn^2}{64} + \frac{cn^2}{64} + \frac{cn^2}{16} = \frac{25cn^2}{256}$

$T(n) = c\left(n^2 + \frac{5n^2}{16} + \frac{25cn^2}{256} + \cdots\right)$

here, $a = \frac{5}{16}$, $Sn = \frac{1}{1-r}$

$$T(n) = Cn^2\left(1 + \frac{5}{16} + \frac{25}{256} + \cdots\right)$$

$$= Cn^2\left(\frac{1}{1-5/16}\right) = Cn^2 \cdot \frac{16}{11}$$

$$\boxed{TC = O(n^2)}$$

Q(5) : what is time complexity of

```
int fun(int n){
    for(int i=1; i<=n; i++){
        for(int j=1; j<n; j+=i){
            // Some O(1) task
}}}
```

i	j	time
1	1 to n	n-1
2	1 to n	(n-1)/2
3	1 to n	(n-1)/3
⋮	⋮	⋮
n	1 → n	(n-1)/n
		n log n

$$\boxed{TC = O(n \log n)}$$

Q(6):- what should be time complexity

```
for(int i=2; i<n; i = pow(i,k)){
    {
        //O(1) task)
    }
```

where, k is a constant.

$i = 2, 2^k, 2^{k^2}, 2^{k^3} \cdots 2^{k^x}$

$n = 2^{k^x}$

$\log n = k^x \log 2$

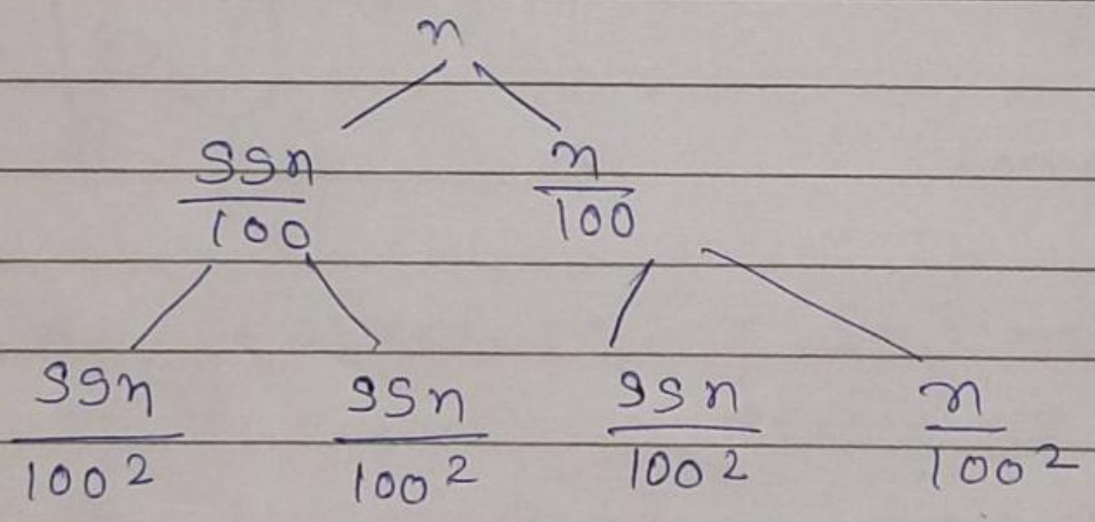
$\log \log n = x \log k$

$\frac{\log \log n}{\log 2}$

$x = \frac{\log \log n}{\log 2 * \log k}$

$\boxed{TC = O(\log \log n)}$

Q(7) Write recurrence relation when quick sort repeadly divides the array into two parts of 99% and 1%. Drive the time complexity in this case. Show the recursion tree while deriving time complexity and find the difference in heights of both extreme parts. What do you understand by this analysis?



taking longer branch that is = $\frac{99n}{100}$

$Tc = \log \frac{100n}{99} \cong \log n$

$n = \left(\frac{99}{100}\right)^k$ or $k = \log\left(\frac{100n}{99}\right)$

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

M T W T F S S

Page No.:	YOUVA
Date:	

Q 8. Increasing order of rate of growth.

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

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

(c) $96 < \log_8 n < \log_2 n < 5n < n\log_2 n < n\log_2 n < 8n^2 < 7n^3 < \log n! < 8^{2n} < 8^n < n!$