Name - Anshul Kumar
Sec - CST SPL-2
Roll no - 11

classmate Date Page

Tutorial - 2

Ans 1. values after execution of while loop,

1st time $i = 1$

2nd time $i = 3$ $(i = 1+2)$

3rd iteration $i = 6$ $(i = 1+2+3)$

4th $i = 10$ $(i = 1+2+3+4)$

let for i^{th} time,

$$i = (1+2+3+ \cdots +i) < n$$

$$= i(i+1)/2 < n$$

$$= i^2 < n$$

$$i = \sqrt{n}$$

therefore, time complexity = $O(\sqrt{n})$

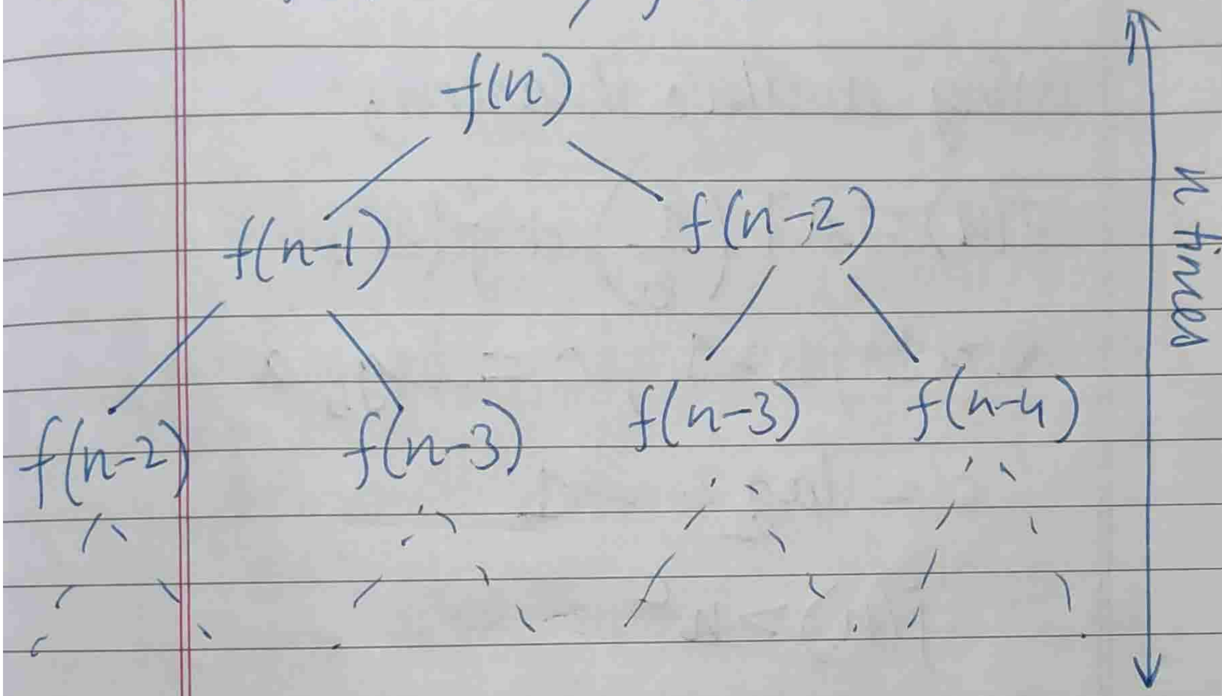
Ans

ANSHUL

Ans 2. For Fibonacci Series:

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

$$f(0) = 0 \quad , \quad f(1) = 1$$



so at every call of function, we get two more function call.

for n levels, $2 * 2 * 2 * \cdots$ (n times)

$$\therefore \underline{T(n) = O(2^n)}$$

Incase of max. no of calls, Space Complexity $= O(n)$
(for recursive calls)

Without recursive stack, Space Complexity $= O(1)$

ANSHUL

classmate

Date

Page

Ans 3.

(i) n(logn)

```
for(i=1; i<=n; i*=2){
    for(j=1; j<=n; j++)
        sum+=i;
}
```

(ii) n^3

```
for(i=1; i<=n; i++){
  for(j=1; j<=n; j++){
    for(k=1; k<=n; k++)
      cout<< i+j+k;
  }
}
```

(iii) log(logn)

```
for(i=2; i<n; i=i*i)
    cout<<i;
```

ANSWL

Ans4. assuming $T\left(\frac{n}{2}\right) \geq T\left(\frac{n}{4}\right)$

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

using master's theorem,

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$a \geq 1$, $b > 1$, $c = \log_b a$

$$c = \log_2 2 = 1$$

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

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

$$\underline{T(n) = \theta(n^2)}.$$

ANSHUL

Ans 5. Inner loop depends on i,

checking no. of iterations of inner loops for each value of i,

$$= \frac{n-1}{1} + \frac{n-1}{2} + \frac{n-1}{3} + \cdots + \frac{n-1}{n-1} + 1$$

$$= \frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \cdots + \frac{n}{n-1} - \log(n-1)$$

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

$$= n\log(n-1) - \log(n-1)$$

$$= n\log(n-1)$$

$$= n\log n$$

Time Complexity $= O(n(\log n))$

NSHUL

Ans 6. for ($i=2$; $i<=n$; $i = pow(i, k)$)
{
// $O(1)$
}

$i = 2^1$
$i = 2^k$
$i = 2^{k^2}$
$i = 2^{k^3}$
$i = 2^{k^4}$
⋮
$i = 2^{k^m}$

$\therefore 2^{k^m} <= n$

$k^m = \log_2(n)$

$m = \log_k(\log_2 n)$

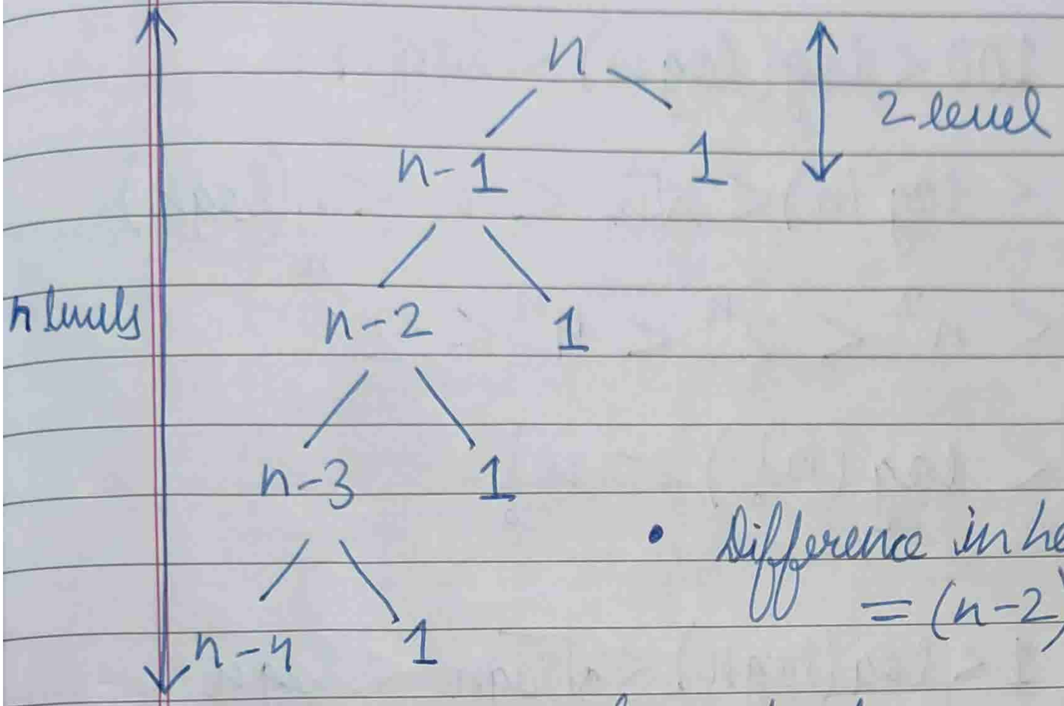
and $O(1)$ for m times.

$\therefore T(n) = O(\log_k(\log_2 n)) + \cancel{O(1)}$

Time Complexity $= O(\log_k(\log_2 n))$

ANSHUL

Ans7. $T(n) = T(n-1) + O(1)$



• Difference in heights $= (n-2)$

lowest height $= 2$
maximum height $= n$

$$T(n) = [T(n-1) + T(n-2) + \cdots + T(1) + O(1)] * n$$

$$= n * n = n^2$$

Time complexity $= O(n^2)$

We understand that when divided into two parts of 99:1 proportion, the time complexity is very high/worst which comes out to be $O(n^2)$.

If they were divided in equal proportion then we could have got better/best time complexities which is as low as $O(n \log n)$.

classmate

Date

Page

Ans 8.

(a) $100 < \log(\log n) < \log(n)$

$< \log^2(n) < \sqrt{n} < n < n(\log n)$

$< n^2 < 2^n < 4^n < 2^{2^n}$

$< \log(n!) < n!$

(b) $1 < \log(\log n) < \sqrt{\log n} < \log n$

$< \log 2n < 2 \log n < n < 2n$

$< 4n < n(\log n) < n^2$

$< \log(n!) < n! < 2(2^n)$

(c) $96 < \log_8(n) < \log_2(n) < 5n$

$< n(\log_6 n) < n(\log_2 n) < n!$

$< \log n! < 8^{2n}$

ANSHUL