Name - Dheeraj

Section - CST

Roll No. - 23

Subject - DAA

Tutorial - 2

Dheeraj

Ques 1 ⇒ What is the time Complexity of below Code and how?

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

Ans ⇒ Time Complexity - $O(\sqrt{n})$ or $O(sqrt\,n)$

for $j=1$ $\quad i=1$

$j=2$ $\quad i=1+2$

$j=3$ $\quad i=1+2+3$

⋮

$j=n$ $\quad i = 1+2+3+\underset{\text{upto } x}{\dots} < n$

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

$x^2 < n$ (ignore Constant)

$x = \sqrt{n}$ or $x = sqrt(n)$

Dhoraj

Ques 2 ⇒ Write recurrence relation for the recursive function that prints fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity of this program and why?

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

Code for fib(n) :-
```
int fib(int n)
{ if (n<=1)
      return 1;                          — O(1)
  return fib(n-1) + fib(n-2);            - T(n-1) + T(n-2)
}
```

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

Recursive tree ⇒

fib(n) → 1

fib(n-1)   fib(n-2) → 2

fib(n-2)   fib(n-3)   fib(n-3)   fib(n-4) → 4

fib(n-3) → 8

⋮

2^n

$1 + 2 + 4 + 8 + \dots$

$a = 1, r = 2$ for $a > 1$, $a\left(\dfrac{r^{terms} - 1}{r - 1}\right)$

$\Rightarrow 1\left(\dfrac{2^{n+1} - 1}{2 - 1}\right) = 2^{n+1} - 1$ {neglect Constant}

$\approx 2^n$

Time Complexity → $\underline{O(2^n)}$

Space Complexity => Space Complexity is depend on the maximum depth of the recursive tree.

So, Space Complexity is $O(n)$

Ques 3 => Write programs which have Complexity $n(\log n)$, n^3, $\log(\log n)$

Ans => (i) $O(n\log n)$

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

(ii) $O(n^3)$

```
for( i=0; i<n; i++)
{
  for(j=0; j<m; j++)
  {
    for(k=0; k<m; k++)
    m[i][j] = a[i][k] * b[k][j];
  }
}
```

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

Dheeraj

```
for(i=2; i<n; i= i*i)
{
    c++;
}
```

Ques4 ⇒ Solve the following recurrence relation

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

Ans ⇒

$$T(n) = 2T\left(\frac{n}{2}\right) + cn^2 \qquad T\left(\frac{n}{2}\right) \geq T\left(\frac{n}{4}\right)$$

Using master's method,

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

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

$a = 2$, $b = 2$

Now $c = \log_2 2 = 1$

Now Compare n^c & $f(n)$

$n^c \to n$

$f(n) = n^2$

As $f(n) > n^c$

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

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

Scanned By ScanIt

Dheeraj

Ques 5 ⇒ What is the time Complexity of following Functions fun()?

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

Ans ⇒ for i=1, j = 1, 2, 3, 4 --- n (for n times)

i=2, j = 1, 3, 5 .... (for n/2 times)

i=3, j = 1, 4, 7 ..... (for n/3 times)

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \cdots$$

$$n\left[1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}\right]$$

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

$$T(n) = n\log n$$

Dheeraj

Ques 6 ⇒ What should be the time Complexity of

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

Where k is a Constant

Ans ⇒ for first iteration, $i = 2$

" 2nd " , $i = 2^k$

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

⋮

for nth iteration, $i = 2^{k^j}$

where, $2^{k^j} <= n$

Apply log both side,

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

$k^j = \log n$

Again Apply log both Side,

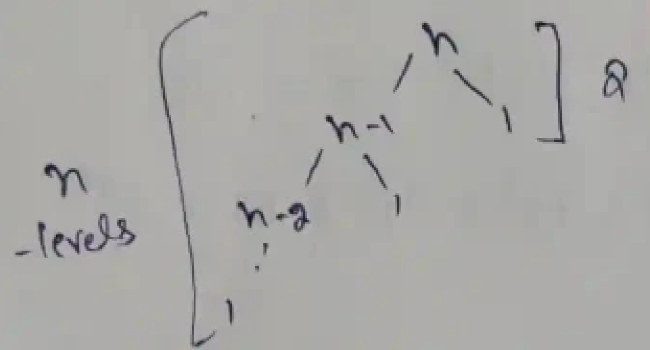
$\log k^j = \log \log (n)$

$j \log k = \log \log (n)$

$j = \log_k (\log n)$

Q7 ⇒ Write a recurrence relation when quick sort repeatedly divides the array in two parts of 99% & 1%. Derive the time complexity in this case. Show the ~~recurrence~~ recursion tree while deriving time complexity and find the difference in heights of both the extreme parts. What do you understand by this analysis?

Ans ⇒ $\therefore T(n) = T(n-1) + O(1)$



'n' work is done at each level for merging

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

$$= n \times n$$

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

Lowest height $= 2$

height height $= n$

diff $= n-2$ $\quad n > 1$

The given algorithm produces linear result.

Ques 8 Arrange the following in increasing order of rate of growth...

Dheeraj

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

(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) $96 < \log_8(n) < \log_2(n) < 5n < n\log_6 n$
$< n\log_2 n < \log(n!) < 8n^2 < 7n^3 < n! < 8^{2n}$