Satyam Rawat
2017006
G

7

TUTORIAL 2

Q1.) Find time complexity:

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

Soln.)

i	j
0	1
1	2
1+2	3
1+2+3	4

$\Rightarrow$ kth Term is of form $= \frac{k(k+1)}{2}$

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

$\Rightarrow k^2 > n$

$k > \sqrt{n} = O(\sqrt{n})$

Q2.) Write recurrence relation & function to print Fibonacci Series. Solve & get complexity. Also find space complexity.

Soln

```
int fibo(int n)
{ if (n<=1) return n;
   return fib(n-1) + fib(n-2);
}
```

$\Rightarrow T(n) = T(n-1) + T(n-2)$, $T(0) = T(1) = 1$

Let $T(n-1) = T(n-2)$ Lower Bound

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

$= 2^2 (T(n-2\times2))$

$= 2^3 (T(n-2\times3))$

$= 2^k (T(n-2k))$ $\quad n-2k = 0 \Rightarrow k = \frac{n}{2}$

$= 2^{n/2}\, T(1) = 2^{n/2} = O(2^n)$

My $T(n-2) \simeq T(n-1)$ (upper bound)

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

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

$= 2^k T(n-k)$

Let $n-k=0 \Rightarrow k=n$

$\Rightarrow T(n) = 2^n\, T(0)$

$= 2^n = O(2^n)$

Space complexity - $O(n)$

Q3) Write porgrams with complexity:

(a) $n\log n$

```
for (int i=0 ; i<n ; i++)
   for (int j=1; j<=n; j*=2)
      cout << "Hii";
```

(b) n^3

```
for (int i=0 ; i<n ; i++)
   for(int j=0 ; j<n; j++)
      for (int k=0 ; k<n; k++)
         cout << "Hi";
```

(c) $\log(\log n)$

```
for(int i=1; i<=n ; i*=2)
   for(int j=1 ; j<=n; j*=2)
      cout << "Hi";
```

Scanned with CamScanner

Q4 Solve:

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

Let $T\left(\frac{n}{2}\right) >= T\left(\frac{n}{4}\right)$

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

Applying master's theorem,

$a = 2 \quad b = 2 \quad f(n) = cn^2$

$n^{\log_b a} = n^{\log_2 2} = n$

$\therefore n < n^2$

$\Rightarrow T(n) = \Theta(n^2)$

Q5 Find time complexity:

```
int fun (int n)
{ for (i=1 to n, i++)
    for (j=1 to n, j+=i)
      { }
}
```

$i=1 \quad j = 1, 2, 3, 4. \quad n =$ n times

$i=2 \quad j = 1, 3, 5 \ldots \quad n = \frac{n+1}{2}$

$i=3 \quad j = 1, 4, 7 \ldots n \quad = \frac{n+2}{3}$

$i=4 \quad = \frac{n}{4}$ times

$i=n \quad = \frac{n}{n}$ times

Hence total time complexity

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

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

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

Q6.) for (int i = 2; i <= n; i = pow (i, k))

{

}

Find T. Complexity.

$i = 2, 2^k, (2^k)^k, ((2^k)^k)^k \ldots$
$= 2, 2^k, 2^{k^2}, 2^{k^3} \ldots$

Loop will end if $i > n$

$\Rightarrow 2^{k^m} > n$

$k^m \log_2 2 > \log_2 n$

$k^m > \log_2 n$

$m \log k > \log(\log n)$

$m > \frac{\log(\log n)}{\log(k)} \rightarrow$ const.

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

Q7> Write a recurrence relation when Quick sort repeatedly divides the array into 2 parts of 99% & 1%, Derive T.C., Show recursion tree & find the difference in heights of both extreme parts. What do you understand by this analysis?

Soln Dividing 99% to 1% is worst case of QuickSort

Recurrence relation $\rightarrow$

$T(n) = T(0) + T(n-1) + \theta(n)$

$T(n) \simeq T(n-1) + \theta(n)$

let $\theta(n) = cn$

Applying ~~Masters~~, Backward substitution,

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

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

$T(n) = T(n-3) + c(n-2) + c(n-1) + cn$

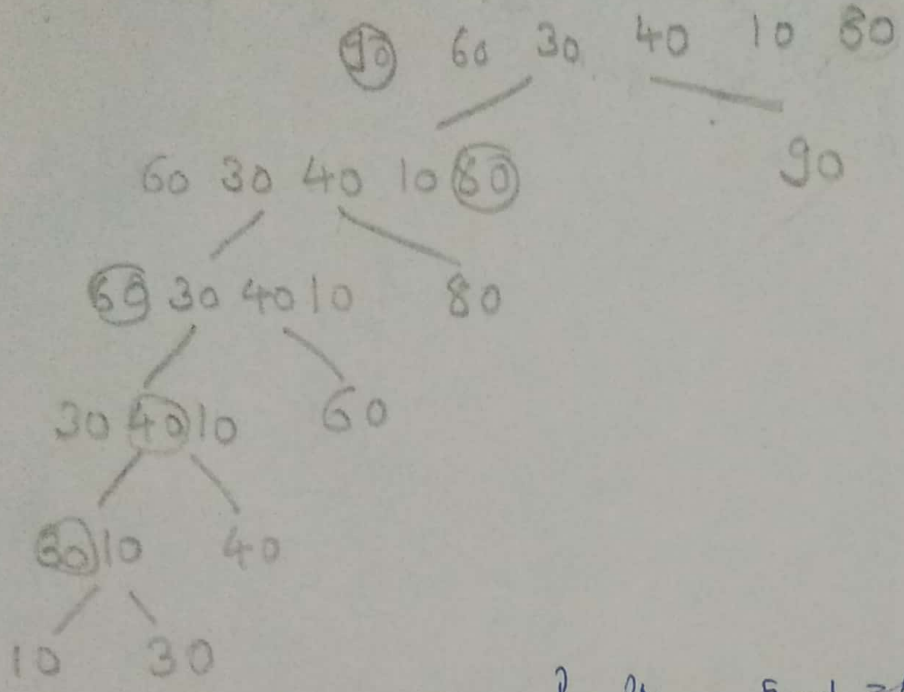
$= T(n-3) + 3cn - c(2+1+0)$

$T(n) = T(n-k) + kcn - c(k-1)$

Put $n-k=1$

$T(n) = T(1) + k(n-1)nc - c(n-2) = kn^2c - knc - cn + C_2$

$= O(n^2)$



Difference in height $= 5-1 = 4$

We understand that for worst case Quicksort has complexity of $O(n^2)$.

Q8) Arrange in Increasing order :

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

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

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

$1 < \log(\log n) < \sqrt{\log n} < \log n < \log 2n < 2\log n < n < \log n! < 2n < 4n < n\log n < n^2 < 2.2^n < n!$

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

Soln 96, $\log_8 n$, $\log_2 n$, $n\log_6 n$, $n\log_2 n$, ~~$n\log_2 n$~~, $\log(n!)$
, $5n$, $8n^3$, $7n^3$, $n!$, 8^{2n}.

Scanned with CamScanner