Tutorial-2

①

Aditya Thapliyal
F - 40
2016593

Q1) $j=1$ $\quad i=1$
$j=2$ $\quad i=1+2$
$i=1+2+3$] m-level

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

$$\frac{m(m+1)}{2} < n$$
$$m \approx \sqrt{n} \qquad \text{Hence } T(n) = \sqrt{n}$$

Q2) Recurrence relation for function that prints Fibonacci series.

$$f(n) = f(n-1) + f(n-2) \qquad f(0)=0,\ f(1)=1$$

$f(n)$
$f(n-1)$ $\quad f(n-2)$
$f(n-2)$ $\quad f(n-3)$ $\quad f(n-3)$ $\quad f(n-4)$

For every function call we get 2 function calls for n levels

so $2 \times 2 \times 2 \cdots$ n times
$T(n) = 2^n$

Max space : Considering recursive stack: no of calls $= n$
Each call has space complexity $O(1)$
$\therefore T(n) = O(n)$

without considering recursive stack
$T(n) = O(1)$

3) write programs with complexity $n(\log n), n^3, \log(\log n)$

1)
```
for (int i=0 ; i<n : ++i)
{ for( int j=1 ; j<=n ; j*=2)
    { sum += 1;
    }
}
```
$T(n) = O(n \log n)$

2) n^3

```
for (int i=0 ; i<n ; ++i)
for (int j=0 ; j<n ; ++j)
for (int k=0 ; k<n ; ++k)
{ sum++; }
```

(2.2)

(3) $\log(\log n)$

```
for ( int i = ; i<=n ; i*= )
  {   sum++;
  }
```

4) solve.

$T(n) = T(n/4) + T(n/2) + Cn^2$

n — 0

$T(n/4)$ $T(n/2)$ — 1

$T(n/3)$ $T(n/8)$ $T(n/4)$ $T(n/8)$ — 2

$0 \to Cn^2$ $\quad 1 \to \frac{n^2}{4^2} + \frac{n^2}{2^2} = \frac{C5n^2}{16}$

$2 \to \frac{n^2}{8^2} + \frac{n^2}{16^2} + \frac{n^2}{4^2} + \frac{n^2}{8^2} = \left(\frac{5}{16}\right)^2 n^2 C$

max level $= \frac{n}{2^k} = 1 \quad = k = \log_2 n$

$T(n) = C n^2 + \left(\frac{5}{16}\right) n^2 + \left(\frac{5}{16}\right)^2 n^2 + \dots + \left(\frac{5}{16}\right)^{\log n} n^2)$

$T(n) = Cn^2 \left[1 + \left(\frac{5}{16}\right) + \left(\frac{5}{16}\right)^2 + \dots + \left(\frac{5}{16}\right)^{\log n} \right]$

$T(n) = Cn^2 \times + \times \frac{\left(1 - (5/16)^{\log n}\right)}{1 - 5/16}$

$T(n) = Cn^2 \times \frac{11}{5} \times \left(1 - \left(\frac{5}{16}\right)^{\log n}\right)$

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

$O(Cn^2)$ Ans.

(6) What's the time complexity of fun()

for i j

1 1

2 2 4 3 6 5 j = (n-1)/i times

3 3 6 9

4 8 12

$\sum_{i=1}^{n} \frac{(n-1)}{i}$ $\therefore T(n) = \frac{(n-1)}{1} + \frac{(n-1)}{2} + \frac{(n-2)}{3} + \frac{(n-1)}{n}$

$T(n) = n[1 + \frac{1}{2} + \frac{1}{3} + \dots \frac{1}{n}] - 1 \cdot [1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}] = n\log n - \log$

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

(Q 8) i

= 1 2

2 k 2^k

3 2^k k $(2^k)^k = 2k^2$

$2^{2^{2k}}$ 2km

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

2 $\therefore \sum_{i=1}^{m}$ — $k^m = \log_2 n$ 1+1+1 .... m times

$m = \log_k \log_2 n$ $T(n) = O(\log_k \log n)$

2^{k^2} $(2^k)^{k^k}$

(7) Write a recurrence relation when quick sort repeatedly divides array into two parts of 99 % & 1 %. Derive time complexity in this case.

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

n levels [n ; n-1 1 ; n-2 ; 1]

$T(n) \{ T(n-1) + T(n-2) + \dots T(1) + O(1) \}$ $= n \times n$

$\therefore T(n) = O(n^2)$

lowest height = 2

highest = n

diff = n-2 n>1

Hence linear result.

(23)

Aditya

Arrang in inc order of rate of growth. (2-4) Adityo

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

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

(c) $\log_3 n < \log 2n < 5n < n\log(n) < n\log_2 n < \log(n!) < 8n^2 < 7n^3 < n! < 8^{2n}$