

Asitika Thapliyal  
F-40  
2016593

Tutorial-2

①

(a)  $\begin{matrix} j=1 & i=1 \\ j=2 & i=1+2 \\ & i=1+2+3 \end{matrix} \quad \left. \vphantom{\begin{matrix} j=1 \\ j=2 \end{matrix}} \right\} \text{m-level}$

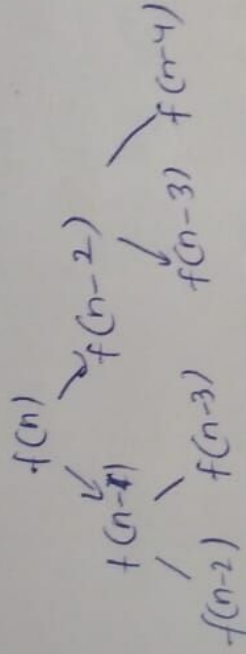
For (i) :- 1 + 2 + 3 + ... + n

$$\begin{aligned} m(m+1) &< n \\ \frac{m^2}{2} &\approx \sqrt{n} \end{aligned}$$

Here  $T(n) = \sqrt{n}$

Q2) Recurrence relation for function that prints Fibonacci series.

$$f(n) = f(n-1) + f(n-2) \quad \begin{matrix} f(0)=0 \\ f(1)=1 \end{matrix}$$



For every function call we get 2 function calls for  $n$  words

So  $2 \times 2 \times 2 \times \dots \times n$  times

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

Max space : Considering recursive

Stack: no of calls = n

FEAN Each call has space complexity  $O(1)$   
∴  $T(n) = O(n)$

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)  $n^3$

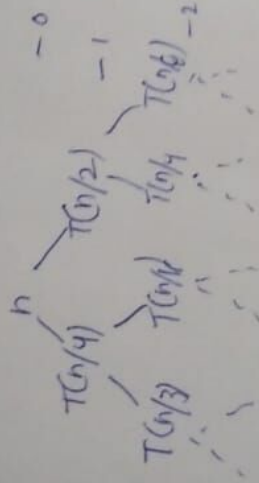
```
for (int i=0; i<n; i++)
  for (int j=0; j<n; j++)
    for (int k=0; k<n; k++)
      {
        // sum++
      }
```

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

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

4) same.

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



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

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

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

$$T(n) = Cn^2 + \left(\frac{5}{16}\right)n^2 + \left(\frac{5}{16}\right)^2 n^2 + \dots + \left(\frac{5}{16}\right)^{\log_2 n} n^2$$

$$T(n) = Cn^2 \left[ 1 + \left(\frac{5}{16}\right) + \left(\frac{5}{16}\right)^2 + \dots + \left(\frac{5}{16}\right)^{\log_2 n} \right]$$

$$T(n) = Cn^2 \times \left( 1 - \left(\frac{5}{16}\right)^{\log_2 n} \right) \frac{1 - \frac{5}{16}}{1 - \frac{5}{16}}$$

$$T(n) = Cn^2 \times \frac{1}{5} \times (1 - \left(\frac{5}{16}\right)^{\log_2 n})$$

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

23  
Ans.

Q1) What's the time complexity of  $\text{fun}(1)$

fun(1)   
 1   
 2   
 3   
 4   
 5   
 6   
 7   
 8   
 9   
 10   
 11   
 12   
 13   
 14   
 15   
 16   
 17   
 18   
 19   
 20   
 21   
 22   
 23   
 24   
 25   
 26   
 27   
 28   
 29   
 30   
 31   
 32   
 33   
 34   
 35   
 36   
 37   
 38   
 39   
 40   
 41   
 42   
 43   
 44   
 45   
 46   
 47   
 48   
 49   
 50   
 51   
 52   
 53   
 54   
 55   
 56   
 57   
 58   
 59   
 60   
 61   
 62   
 63   
 64   
 65   
 66   
 67   
 68   
 69   
 70   
 71   
 72   
 73   
 74   
 75   
 76   
 77   
 78   
 79   
 80   
 81   
 82   
 83   
 84   
 85   
 86   
 87   
 88   
 89   
 90   
 91   
 92   
 93   
 94   
 95   
 96   
 97   
 98   
 99   
 100

$$T(n) = T(n-1) + (n-1) + \frac{n}{2} + \frac{(n-1)}{2}$$

$$T(n) = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = O(n^2)$$

$$T(n) = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} = O(n^2)$$

$$2^k m \leq n$$

$$k m = \log_2 n$$

$$m = \log_2 \log_2 n$$

$$1 + 1 + \dots + m \text{ times}$$

$$T(n) = O(k \log_2 n)$$

(\*) write a recursive relation when quick sort repeatedly divides array into two parts of  $\frac{n}{2}$  &  $\frac{n}{2}$ . Derive time complexity in this case.

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

$$T(n) = T(n-1) + T(n-2) + \dots + T(1) + O(1) = O(n^2)$$

Lowest height = 2  
heights  $\rightarrow \rightarrow \rightarrow n$

$$\text{diff} = n-2 \quad n > 1$$

Here linear result.



Arrange in inc order of growth.

Q-1

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

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

$$(b) \quad 2 \log n, 4 \quad 1 < \log \log n < \sqrt{\log n} < \log n < \log 2n < 2 \log n < n < n \log n$$

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

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