Name - Samanvidhi
Section - F
Roll no. - 24
University Roll no. - 2016985

Date

Tutorial - 2

1. $i = 0, 1, 3, 6, 10, 15, 21 \;--\; n$

let the sum of above K terms is S_K

$$S_K = 1+3+6+10+ \;---\; + T_K \quad —— (1)$$
$$S_{K-1} = 1+3+6+10 \;-----\; + T_{K-1} \quad —— (2)$$

Subtract (2) from (1)

$$T_K = S_K - S_{K-1} = 1+2+3+4 \;--\; + K$$

We have. $T_K = n$

$$\therefore 1+2+3+4 \;---\; + K = n$$

$$\frac{K(K+1)}{2} = n \quad = \quad K^2 + K - 2n = 0$$

$$K = \frac{-1 \pm \sqrt{8n+1}}{2}$$

taking the values we get total no. of times the loop runs for $i = K+1 = \frac{\sqrt{8n-1}}{2}$

$$T.C.\ T(n) = O\left(\frac{\sqrt{2n+1}}{2}\right) = O(\sqrt{n}).$$

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

$$T(n+1) \simeq T(n-2)$$

Mohan

Date

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

$T(n-2) = 2 * (2T(n-2-2) + c) + c$

$= 4T(n-2) + 3c.$

$T(n-4) = 2 * (4T(n-2) + 3c) + c$

$= 8T(n-3) + 7c$

Generalizing.

$= 2^k T(n-k) + (2^k - 1)c$

put $n-k = 0$

$n = k$

put $n = k$

$T(n) = 2^n * T(0) + (2^n - 1)c$

$= 2^n * 1 + 2^n c - c$

$= 2^n (1 + c) - c$

$= 2^n$

Time complexity $= O(2^n)$

☆ Space complexity is proportional to the max. depth of recursion tree.

Hence space complexity of fibbonacci recursion is $O(N)$.

3\. 1. $n(\log n)$

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

Mohan

Date

2. n^2

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

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

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

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

$\therefore T\left(\frac{n}{4}\right) \simeq T\left(\frac{n}{2}\right)$

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

As $a \geq 1$ and $b > 1$
using Master's method.

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

$c = \log_b a$

$c = \log_2 2 = 1$

$f(n) = n^c$

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

Mohan

Scanned by CamScanner

Date

5. for $i=1$, j is $1,2,3,4$ --- run for n times
for $i=2$, j is $1,3,5$ --- upto $n/2$ times
for $i=3$, j is $1,4,7$ --- upto $n/3$ times

$$T(n) = n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + ---$$

$$= n\left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + --\right)$$

$$= n \int_1^n dn/x$$

$$= [\log n]_1^n$$

Time complexity = $n \log n$.

6. for first iteration $i=2$.
second " $i = 2^k$
third " $i = (2^k)^k = 2^{k^2}$

n^{th} iteration $i = 2^k$ loop end at $2^i = n$

apply log $n = \log_2 k^i$

$$k^i = \log n$$

$$i = \log_e(\log n).$$

7. 99 to 1 in quick sort.

Mohan

Scanned by CamScanner

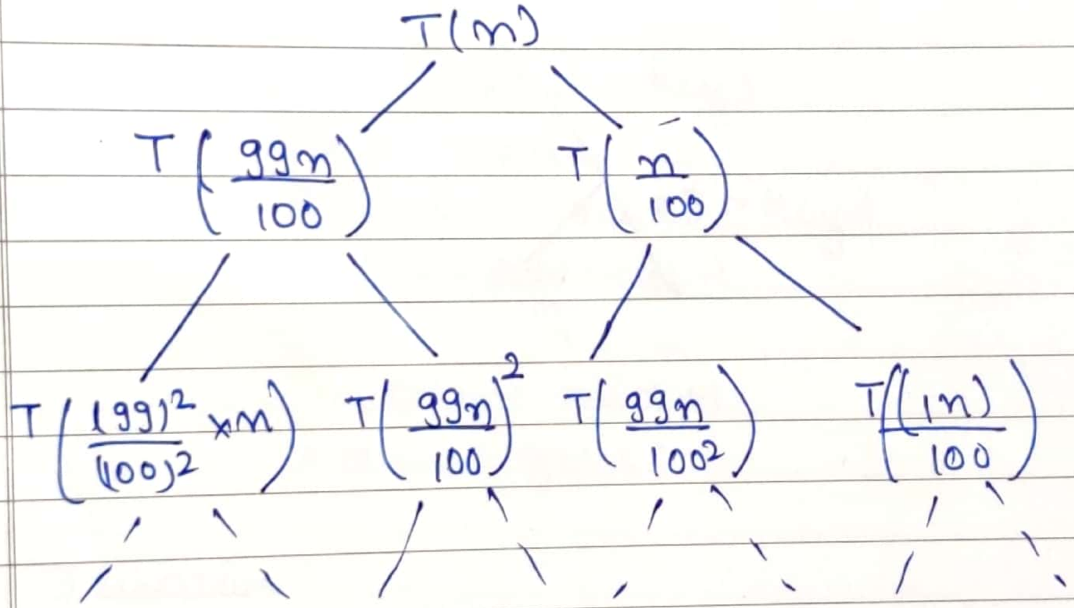
Date

When pivot is when from front or end always.

So,

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right) + O(n)$$

$$T(n) = T\left(\frac{99n}{100}\right) + T\left(\frac{n}{100}\right) + O(n)$$



$$n = \left(\frac{99}{100}\right)^k$$

$$\log n = k \log \frac{99}{100}$$

$$k = \log_{\frac{100}{99}} n$$

$$TC = n * \log_{\frac{100}{99}} (n).$$

Date

8. (a) $100 < \log\log(n) < \log^2 n < \log n <$
$\log n! < n < n\log n < n^2 < 2^n < 4^n$
$< 2^\wedge(2^\wedge n) < n!$

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

Mohan