Nancy Jain
F
64

Tutorial - 2

Q1. What is the time complexity of below code?

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

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

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

$j=3$ $\quad$ $i=\cancel{1}+2+3$

for (i)

$\therefore$ $1+2+3+ \dots + < n$

$\therefore$ $1+2+3+ \dots m < n$

$$\frac{m(m+1)}{2} < n$$

$$m \simeq \sqrt{n}$$

By summation method,

$\Rightarrow \sum_{i=1}^{m} \to 1+1+\dots+\sqrt{n}$ times

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

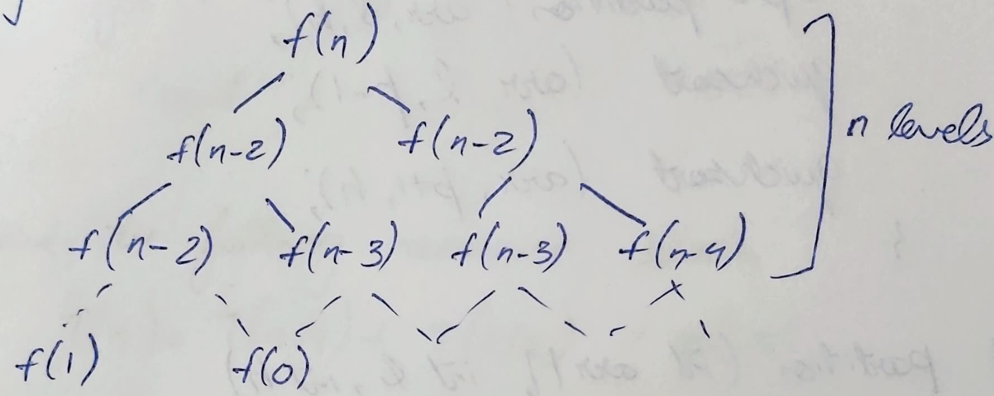
$$T.C = O(\sqrt{n})$$

2. Write recurrence relation for function that prints fibonacci series. Solve it to get the time complexity. What will be the space complexity and why?

For fibonacci series,

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

by



$\therefore$ At every function call we get 2 function calls

$\therefore$ for n levels

we have $= 2 \times 2 \ldots n$ times

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

Space Complexity:

Considering recursive stack:

no. of calls maximum $= n$

For each call we have space complexity $O(1)$

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

without considering recursive stack each call we have time complexity $O(1)$

$\therefore T(n) = O(1)$

3. Write program which have complexity
n (logn), n^3, log (logn)

i) n logn → Quick Sort

```
void quicksort(int arr[], int l, int h)
{
    if(l<h)
    {
        int p= partition(arr, l,h);
        quicksort (arr, l, p-1);
        quicksort (arr, p+1, h);
    }
}
int partition (int arr[], int l, int h)
{
    int pivot = arr [h];
    int i = l-1;
    for (int j = l; j<= h-1 ; j++)
    {
        if(arr[i] < pivot)
        { i++;
        swap (&arr[i], &arr[j]);
        }
    }
```

) n^3 -> Multiplication of 2 square matrix

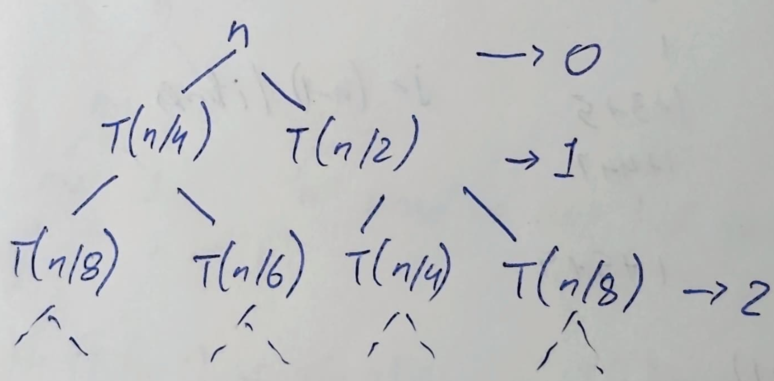
```
for (i=0; i<n1; i++)
  for (j=0; j<c2; j++)
    for (k=0; k<c1; k++)
      res[i][j] += arr[i][k] * b[k][j];
```

iii) log(logn)

```
for (i=2; i<n; i=i*i)
    count++;
```

4. Solve the following recursive relation.

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



At level :

$0 \rightarrow Cn^2$

$1 \rightarrow \frac{n^2}{4} + \frac{n^2}{2^2} = \frac{C5n^2}{16}$

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

!

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

$K = \log_2 n$

$$T(n) = C\left(n^2 + \left(\frac{5}{16}\right)n^2 + \left(\frac{5}{16}\right)^2 n^2 + \dots + \left(\frac{5}{16}\right)^{\log_4 n} n^2\right)$$

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

$$= Cn^2 \times 1 \times \left(\frac{1-(5/16)^{\log_4}}{1-(5/16)}\right)$$

$$T(n) = n^2 \times \frac{11}{5} \times \left(1 - \left(\frac{15}{16}\right)^{\log n}\right)$$

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

$$T(n) = O(n2)$$

5. for

i	j
1	1
2	1+3+5
3	1+4+7
⋮	⋮
n	1+5+9

$j = (n-1)/i$ times

$$\sum_{i=1}^{n} \left(\frac{n-1}{i}\right)$$

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

$$T(n) = n\left[1 + 1/2 + 1/3 + \dots + \frac{1}{n}\right]$$

$$- 1 \times \left[1 + 1/2 + 1/3 \dots + \frac{1}{n}\right]$$

$$= n \log n - \log n$$

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

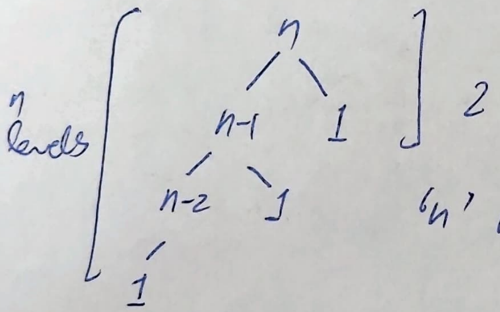
	i	where
for	2^1	
	2^k	$2^k <= n$
	2^{k^2}	$k^m = \log_2 n$
	2^{k^3}	$m = \log_k \log_2 n$
	$\vdots$	
	2^{k^n}	

$\therefore \sum_{i=1}^{m} 1$ $\quad 1+1+1 \text{ --- } m \text{ times}$

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

7. given algorithm divides array in 99% and 1% part.

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



'n' word is done at each level.

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

$= n \times n$

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

Lowest height $= 2$, highest height $= n$

$\therefore$ difference $= n - 2$ $\qquad n > 1$

The given algorithm produces linear result.

8.

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

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

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