Nikhil Aswal_ Sec-f_17_ 2016877

Tutorial - 1

Ans. 1. Asymptopic Notation :- Asymptatic notation are the mathematical notations used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value.

big O, big θ, big Ω are the different types of asymptopic notation.

Ans. 2.

2^0 $i=1$

2^1 $i=2$

2^2 $i=4$

2^3 $i=8$

2^4 $i=16$ - - - - - 2^k (K times) for n values

So. $2^k = n$

$\log 2^k = \log n$

$k \log_2 2 = \log_2 n$

$k = \log_2 n$

Hence the time complexity is $O(\log n)$

Ans. 3. $T(n) = 3T(n-1)$ $T(0) = 1$

Let $n = n-1$

$T(n-1) = 3T(n-1-1)$

$T(n-1) = 3T(n-2)$

$T(n) = 3[3T(n-2)]$ —— (2)

$T(n-2) = 3T(n-2-1)$

$T(n) = 3[3 \cdot 3T(n-3)]$ —— (3)

So, from above 3 eq^n we should obtains a Relation

$T(n) = 3^k T(n-k)$

Let $n-k=0$

$n=k$

$T(n) = 3^k T(0)$ Here $T(0)=1$

So, $T(n) = 3^k \cdot 1$

$= 3^n$

So time complexity is $3^n = O(3^n)$

Ans. 4. $T(n) = 2T(n-1)-1$ —— (1) $T(0)=1$

Let $n = n-1$

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

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

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

$T(n) = 4T(n-2)-3$ —— (2)

$n=n-2$

$T(n-2) = 2[2T(n-3)-1]$

$T(n) = 4[2T(n-3)-1]-3$

$T(n) = 8T(n-3)-7$ —— (3)

⋮

$T(n) = 2^k T(n-k) - \{+2^{k-1} + 2^{k-2} + \ldots \ldots 2^2 + 2 + 1\}$

Let $n-k=0$

$n=k$

$= 2^n T(0) 1 - \{1 + 2 + 2^2 + \dots + 2^{k-1}\}$

$= 2^n \times 1 + 2^k + 1$

$= 2^n + 2^n + 1$

$= 2^{n+1}$

$O(2^n)$ is the given time complexity for given relation.

Ans. 5. Here $S_i = S_{i-1} + i$

the value of i increasi by 1 for each iteration the value contained in 's' at the ith iteration is the same sum of the first i positive integer.

If k is the total no. of iteration taken by program

then loop like

$1 + 2 + 3 + \dots + k$

$= \frac{k(k+1)}{2} > n$

So, $k = O(\sqrt{n})$

Hence the time complexity is $O(\sqrt{n})$

Ans. 6.

at	Passes	Let n=16
$i=1$	$k=1$	1
$i=2$	$k=2$	4
$i=3$	$k=3$	9
$i=4$	$k=4$	11
⋮		⋮
$i=n$	for $k=n$	n2

~~14~~ 1 4 9 16 ..... n^2

$$= \frac{n(n+1)(2n+1)}{6}$$

$$= o(\log^2 1) + o(\log^2 2) + \ldots O(\log^2 n) \le c \cdot o(\log^2 n)$$

Hence fore time complexity is $O(\log^2 n)$

Ans. 7

i	j	K	
6	1	1	
7	2	2	
8	4	4	
9	8	8	
10	16	16	(out of bound

So, $\left(\frac{n}{2}\right) \times (\log n) \times (\log n)$

as constants can be ignored,

Here for each value of i it iterates & check the written condition for k,

So,

time complexity is like

$(n \cdot \log n \cdot \log n)$

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

Ans 8.

i	j	no. of times P
1	1	-1
2	2	-2
1	1	1
1	1	1
n	n	n times

P = n times

j = n times

$i \cdot j = = n \cdot n.$

(n) (n) tim.

Here $n = n-3$

$= (n-3)(n-3)$

$O(n^2+9+6n)$

$= \theta(n^2)$ is time complexity.

Ans. 9. Let $n = 12$

```
for (i=1 to n) {
    for (j=1; j<=n; j=j+1)
        Print f ("*");
```

$i=1, \quad j=2,3,4,5, \ldots \ldots (n-1)$

$i=2, \quad j=3,4,5,6 \ldots \ldots (n-1)$

$i=3, \quad j=4,5, \ldots \ldots (n-1)$

$i=n, \quad j=(n+1). \ldots \ldots (n-1)$

for each of i value e.n it iterates through $(n-1)$ times

for $n(n-1)$ times

$= (n^2-n)$

$= O(n^2)$

Hence the time complexity is $O(n \log n)$.