Name -> Mridul Gupta
Section - H

U. Roll No. - 2016852
Roll No. - 38

Tutorial - 1 (DAA)

Ans 1

Asymptotic Notations: are the mathematical notations used to describe the running time of an algorithm:

Different types of Asymptotic Notations:

1) Big-O Notations (O) - It represents of the tight upper bound of algorithm

$f(n) = O(g(n))$ if $f(n) \le c^* g(n)$

2) Omega Notations (Ω) - It represents the tight lower bound of algorithm

$f(n) = \Omega(g(n))$ if $f(n) \ge c^* g(n)$

3) Theta Notations (θ) - It represents upper and lower bound of algorithm.

$f(n) = \theta(g(n))$ if $c_1 g(n) \le f(n) \le c_2 g(n)$

Ans 2

for (i=1 to n)
{ i = i * 2
}

i = 1
i = 2
i = 4
i = 8
= . . . n

It is forming a GP for nth term

$a_n = ar^{n-1}$ then $a = 1, r = 2$

$n = 2^{k-1}$

taking log both side

$\log n = \log 2^{k-1}$

$\log_2 n = k-1 \log_2 2$ $\{\log_2 2 = 1\}$

$\log_2 n = k-1$

$\boxed{k = \log_2 n + 1}$ $O(\log n)$

Ans 3 $T(n) = 3T(n-1)$ $\{$if $n>0$ otherwise $1\}$
$\{T(0) = 1\}$

Put $n = n-1$

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

$T(n) = 3(3T(n-2)) = 9T(n-2)$

Put $n = n-2$

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

$T(n) = 9(3T(n-3)) = 27T(n-3)$

$=$

$=$

$T(n) = 3^k T(n-k)$ $(n-k=0$
$n=k)$

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

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

$T(n) = O(3^n)$

Ans

4) $T(n) = \{ 2T(n-1) - 1$ if $n > 0$ other $1 \}$

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

Put $n = n-1$

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

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

$T(n) = 4T(n-2) - 3$

Put $n = n-2$

$\cancel{T(n-3) = 2T(n-4) - 1}$

$\cancel{T(n) = 4(2T(n-4) - 1) - 1}$

$\cancel{= 8T(n-5) - 1}$

$\cancel{= 2^k T(}$

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

$\therefore$

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

$= 8T(n-3) - 6 - 1$

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

$\vdots$

$T(n) = 2^k T(n-k) - 7$

we have $T(0) = 1$

$\therefore n - k = 0$

$n = k$

$\therefore T(n) = 2^n T(n-n) - 7$

$2^n T(0) - 7$

$2^n \times 1 - 7 = 2^n - 7$

$\therefore O(2^n)$

Ans 5

```
int i=1, s=1
while (s<=n)
{ i++;
  s=s+i;
  Print("#");
}
```

$i=1$ $\quad s=1$

$i=2$ $\quad s=1+2$

$i=3$ $\quad s=1+2+3$

$\vdots \quad \vdots$

Loop ends when $s>n$

$$1+2+3+4 \dots k > n$$

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

$$k^2 > n$$

$$k > \sqrt{n}$$

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

Ans 6

```
void function (int n) {
    int i, count=0;
    for (i=1; i*i<=n; i++)
        count++;
}
```

$i=1$

$i=2$

$i=3$

$\vdots$

$i=k$

Loop ends when $i*i > n$

$$k*k > n$$

$$k^2 > n$$

$$k > \sqrt{n}$$

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

Ans 7

```
void function( int n)
{
    int i, j, k, count=0;
for (i=n/2; i<=n; i++)
for (j=1; j<=n; j=j*2)
 for (k=1; k<=n; k=k*2)
        count++;
}
```

1st loop $i = \frac{n}{2}$ to n ; i++

$O(\frac{n}{2}) = O(n)$

- 2nd Nested loop $j = 1$ to n ; j * 2

$= O(\log n)$

- 3rd loop $k = 1$ to n $k = k * 2$

$= O(\log n)$

Total complexity $= O(n * \log n * \log n)$

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

Ans

Function (int n)
{
if (n==1) return; — 1
~~for (...)~~
for (int j=1 to n) — n^2
printf ("*");
function (n-3) — T(n-3)

$T(n) = T(n-3) + n^2$ $\quad \underline{T(1) = 1}$

Put $n = n-3$

$T(n-3) = T(n-6) + (n-3)^2$

$T(n) = (T(n-6) + (n-3)^2) + n^2$

Put $n = n-6$

$T(n-6) = T(n-9) + (n-6)^2$

$T(n) = ((T(n-9) + (n-6)^2) + (n-3)^2) + n^2$

$= T(n-3k) + (n-(3(k-1))^2) + (n-(3(k-2))^2) + n^2$

$n = 3k = 1$

$n = 3k+1$ $\quad n-1 = 3k$

~~$T(3k+1-3k) + (n-3(n-1)^2) + (n-3(n-2)^2)$~~

$T(n-n+1) + (n-(3(n-2)^2)) + (n-3(n-1)^2)$

~~T(0)~~ ~~tc~~ of. $T(1)=1$

$\therefore T(n) = O(n^3)$

Ans 9 void function (int n)

{
for (i=1 to n) — n

for (j=1; j<n; j=j+1) — n

print (" ");

So for iteration it will take n^2

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

i=1 — j=1 to n
i=2 — j=1 to n
i=3 — j=1 to n
i=4 — j=1 to n

Ans 10 $f(n) = n^k$ $f_2(n) = c^n$

Asymptotic relation b/w f & f_2 $k>=1$; $c>1$

is Big O i.e $f(n) = O(f_2(n)) = O(c^n)$

is $n^k \leq G * c^n$ [G is som