Date _____ Page No. _____

Tutorial-1 (DAA)

Ans1 Asymptotic Notation: Asymptotic Notation are the mathematical notations used to describe the running time of an algorithm.

Different Types of Asymptotic Notation:

1 Big-o Notation (O): Represents upper bound of algorithm.

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

2 Omega Notation (Ω): Represents lower bound.

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

3 Theta Notation (Θ): Represents upper & lower bound.

$f(n) = \Theta(g(n))$ if $C_1 g(n) \leq f(n) \leq C_2 g(n)$

Date _____ Page No. _____

Ans 2

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

$i = 1$
$i = 2$
$i = 4$
$i = 8$
$i = 16$
$i = n$

It is forming GP,

$a_n = ar^{n-1}$
$n = ar^{k-1}$
$n = 1 \times (2)^{k-1}$
$\log n = \log 2^{k-1}$
$\log n = (k-1)\log 2$
$k = \log n + 1$

$a_n = n$
$r = 2$
$a = 1$

Ans 3 $T(n) = 3T(n-1)$ if $n > 0$, otherwise 1

$T(1) = 3T(0)$
$T(1) = 3 \times 1$
$T(2) = 3T(1) = 3 \times 3 \times 1$
$\vdots$
$T(n) = 3 \times 3 \ldots = 3^n = O(3^n)$

Mohan

Teacher's Signature _____

Date _____ Page No. _____

Ans 4 $T(n) = 2T(n-1) - 1$ if $n > 0$, otherwise 1

$T(0) = 1$

$T(1) = 2T(0) - 1$

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

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

⋮

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

Ans 5

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

$i = 1$ $s = 1$

$i = 2$ $s = i + 2$

⋮ ⋮

Loop ends when $s > n$

$1 + 2 + 3 \ldots k > n$

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

$k^2 > n$

Mohan Teacher's Signature _____

Date _____ Page No. _____

$\Rightarrow k > \sqrt{n}$

$= O(\sqrt{n})$

Ans 6

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

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++;
    }
```

Date_____ Page No._____

1st Loop: $i = \frac{n}{2}$ to n, $i++$

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

2nd Nested Loop: $j = 1$ to n, $j = j * 2$

$j = 1$
$\vdots$
$j = n$

$= O(\log n)$

3rd Nested Loop: $k = 1$ to n, $k = k * 2$

$k = 1$
$\vdots$
$k = n$

$= O(\log n)$

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

Ans 8

```
function (int n)
{
  if (n==1) return;
  for (int i=1 to n)          → 1
  {
    for (int j=1 to n)        → n^2
    {
      printf("*")
    }
  }
  function (n-3)              → T(n-3)
}
```

Mohan

Teacher's Signature_____

Date _____ Page No. _____

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

$T(1) = 1$

$T(4) = 17$

$T(7) = 66$

$T(10) = 166$

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

$= O(n^3)$

Ans9

```
void function(int n)
{
  for(int i=1 to n)                    → n
    {
      for(j=1; j<=n; j=j+1)            → n
        {
          printf("*");
        }
    }
}
```

So, for i upto n, it will take n^2

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

Date _____ Page No. _____

Ans 10 $f_1(n) = n^k$, $f_2(n) = c^n$

Asymptotic ~~Notat~~ relation b/w f_1 & f_2 is Big O

i.e. $f_1(n) = O(f_2(n)) = O(c^n)$

$$n^k \leq G * c^n$$