A-1- Asymptotic notations are used to find the complexity of an algorithm when input is very large.

- Big O(o):

$f(n) = O(g(n))$

iff $f(n) \leq c g(n)$

$\forall n \geq n_0$

for some constant $c > 0$

$g(n)$ is "tight upper bound" of $f(n)$

- Big Omega (Ω):

$f(n) = \Omega(g(n))$

iff $f(n) \geq c g(n)$

$\forall n \geq n_0$

for some constant $c > 0$

$g(n)$ is "tight lower bound" of $f(n)$

- Big Beta (Θ):

$f(n) = \Theta(g(n))$

iff $c_1 g(n) \leq f(n) \leq c_2 g(n)$

$\forall n \geq \max(n_1, n_2)$

for some constant $c_1 > 0$ and $c_2 > 0$

$g(n)$ is both "tight upper bound and lower bound of $f(n)$"

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

1, 2, 4, 8, ---- n

Let k^{th} term = n

$n = 1.(2^{k-1})$

Taking log on both sides

$\log n = k-1 \log_2 2$

$k = 1 + \log n$

$O(1+\log n)$

$O(\log n)$ Ans

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

$n = n-1$ in (I)

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

Put (II) in (I)

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

$n = n-2$ in (I)

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

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

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

$n-k = 0$

$n = k$

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

$= 3^n T(0)$

$= 3^n$

$O(3^n)$ Ans

A-4- $T(n) = 2T(n-1)$ - (I)

$n = n-1$ in (I)

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

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

$n = n-2$ in (I)

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

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

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

$n-k = 0$

$k = n$

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

$= 2^n T(0)$

$= 2^n$

$O(2^n)$ Ans

A-6- void function (int n)

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

(1) (√n) (√n) (√n)

$O(1+\sqrt{n}+\sqrt{n}+\sqrt{n})$

$O(1+3\sqrt{n})$

$O(3\sqrt{n})$

$O(\sqrt{n})$

$O(n^{1/2})$ Ans

A-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++;
}
```

i	j	k
n/2	1	1
		2
		4
		.
		.
		n

n/2 → log n → log n

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

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

A-8=

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

n	i	j
n	1	1 to n
	2	1 to n
	⋮	⋮
	n	1 to n
n-3	⋮	⋮
	1	1 to n
n-6	⋮	⋮
	1	1 to n

$1+4+7+\cdots n$

$n = 1+3(K-1)$

$= 3K-2$

$K = \frac{n+2}{3}$

↓ No. of terms

$\frac{n+2}{6}\left[2+\left[\frac{n-1}{3}\right]\times 3\right]$

$\left[\frac{n+2}{6}[n+1]\right]\times n^2$

$O\left[\frac{(n^2+3n+2)\times n^2}{6}\right]$

$O[n^4]$ Ans

A-9-

```
void function (int n)
{
    for (i=1 to n)(n)
    {
        for (j=1; j<=n; j=j+i)   (n) (n^2)
            printf ("*");  (n^2)
    }
}
```

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

$$O(3n^2 + n)$$

$$O(n^2)$$ Ans