NAME → Vaibhav Sharma
Section → CST-SPL-2
Roll No. → 16

Data & Analysis of Algorithms
Tutorial - 1

1. Asymphotic 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 instant $C > 0$
  $g(n)$ 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~~ Theta (θ)

$$f\,n = \theta(g(n))$$

iff

$$c_1\, g(n) \le f(n) \le c_2\, g(n)$$

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

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

$g(n)$ is both "tight upper bonded & lower bound of $f(n)$".

Q. for $(i = 1 \text{ to } n)$, $\{ i = i * 2; \}$

$$1, 2, 4, 8, \_\_\_\ n$$

let K^{th} term = n

$$n = 1 \cdot (2^{K-1})$$

Taking log on both sides

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

$$K = 1 + \log n$$

$$O(1 + \log n)$$

$$O(\log n) \text{ Ans}$$

3. $T(n) = 3T(n-1)$ — ①

$n = n-1$ in ①

$T(n-1) = 3T(n-2)$ — ②

Put ② in ①

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

$n = n-2$ in ①

$T(n-2) = 3T(n-3)$ ③

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

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

$n-k=0$

$n=k$

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

$= 3^n T(0)$

$= 3^n$

$O(3^n)$ Ans

4.

$T(n) = 2T(n-1)$ — ①

$n = n-1$ in ①

$T(n-1) = 2T(n-2)$ — ⅱ

$T(n) = 4T(n-2)$ — ⅲ

$n = n-2$ in ①

$T(n-2) = 2T(n-3)$ — 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

6 Void function (int n)

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

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

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

$O(3\sqrt{n})$

~~[illegible]~~

$O(\sqrt{n})$

$\underline{O(n^{1/2})}$ 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++;
}
```

i	j	k
n/2	1	1 = 1
		2
		4
		⋮
n/2	logn	n → logn

$O(n/2 \times \log n \times \log n)$

$O(\underline{n(\log n)^2})$

Ans

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
		1
		n
	2	1
		1
	|	n
n-3	n	1
	|	1
	|	n
n-6	|	|
|	|	|
;	|	|

$1 + 4 + 7 + \text{----} \; n$

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

$= 3k - 2$

$k = n+2/3$ no. of terms

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

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

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

$O[n^4]$ Ans

9 Void function (int n)

{

for (i=1 to n) n

{

for (j=1; j<=n; j=j+1) n^2 n^2

printf ("*"); n^2

}

}

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

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

$\underline{O(n^2)}$ Ans

10 As given n^k & c^n

relation b/w n^k & c^n is

$n^k = O(c^n)$

$n^k \le a(c^n)$

$\forall n \ge n_0$ &

constant, $a>0$

for $n_0 = 1$

$c = 2$

$\Rightarrow 1^k \le a^{2 \cdot 1}$

$\Rightarrow n_0 = 1$ & $c=2$.