Name - Adhyan Dhyani
Class - CSE - 4th sem.
Roll no - 20-H

Date. ______
Page No. ______

Tutorial - 1

Q1

Sol^n Asymptotic notations are the mathematical notation used to describe the running time of an algo when a input tends towards a particular value.

Types :-

1) Big O notation : Gives worst case time complexity.
2) Omega notation : Gives best case.
3) Theta notation : Gives avg. case.

For eg. In bubble sort, when it is already sorted, the time taken is the least i.e. the best case and when its reverse order, the time taken is quadratic which is the worst case.

Q2

Sol^n $i = 1, 2, 4, 8 \cdot \cdot \; n$

$a_k = ar^{k-1}$ $\qquad a = 1$

$n = 2^{k-1}$ $\qquad r = 2$

Date.
Page No.

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

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

$k = \log_2 n + 1$

$T(n) = O(\log_2 n)$.

Q3 soln

$T(n) = 3T(n-1)$ — (1) $T(0) = 1$

put $n = n-1$ in eq — (1)

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

put (3) in (1)

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

$= 3^2 T(n-2)$ — (4)

put $n = n-2$ in eq (1)

$T(n-2) = 3T(n-2-1) = 3T(n-3)$ (5)

putting (5) in (4)

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

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

let $n-k = 0$, $n = k$

Date. ______
Page No. ______

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

as $T(0) = 1$ from (2)

$T(n) = 3^n$

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

Q4

$T(n) = 2T(n-1) - T$ — (1)

let $n = n-1$ $T(0) = 1$

$T(n-1) = 2T(n-2) - 1$ — (3)

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

$= 2^2 T(n-2) - 2 - 1$ — (4)

let $n = n-2$ in eq (1)

$T(n-2) = 2T(n-3) - 1$ — (5)

(5) $\rightarrow$ (4)

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

$= 2^3 T(n-3) - 2^2 - 2 - 1$ — (6)

$\hookrightarrow T(n) = 2^k T(n-k) - 2^{k-1} - 2^{k-2} \ldots - 2^0$

let $n - k = 0$

$n = k$

Date.
Page No.

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

from (2),

$$T(n) = 2^n - 2^{n-1} - 2^{n-2} \cdots - 2^0$$

$$2^n - \frac{(1 \times (2^n - 1))}{2-1} \Rightarrow T(n) = (2^n - 2^n + 1)$$

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

Q5 Time complexity :-

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

Sol^n

for $i=1, \; s=1$

$i=2, \; s=1+2$

$i=3, \; s=1+2+3$

,

Sum of n natural numbers

so, $S = \frac{k(k+1)}{2}$

Date.
Page No.

To break out of loop, $s > n$

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

$$\frac{k^2+k}{2} > n$$

$$k^2 > n$$

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

Q6.

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

Sol^n

$i = 1, 2, 3, \ldots \ldots n$

$i^2 = 1, 4, 8 \ldots \ldots n$

$i^2 <= n$ or $i <= \sqrt{n}$

so,

$$a_k = a + (k-1)d$$

$$a = 1, \quad d = 1$$

$$a_k <= \sqrt{n}$$

Date.
Page No.

$$\sqrt{n} = 1 + (k-1) \cdot 1$$

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

Q7

```
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
$n/2$	$(\log_2 n)$	$\log_2 n$
$\vdots$	$\vdots$	
n		
$\left(\frac{n}{2}+1\right)$ times	$\log_2 n$	$\log_2 n$

$$O(i*j*k) = O\left(\left(\frac{n}{2}+1\right) \times \log_2 n \times \log_2 n\right)$$

$$= O\left(\left(\frac{n}{2}+1\right) \times (\log_2 n)^2\right)$$

after remaing
const.

Date.
Page No.

$$T(n) = O\left(n\,(\log_2 n)^2\right)$$

Q8

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

Sol^n

$T(n) = T(n-3) + n^2$ — (1)

$T(1) = 1$ — (2)

let $n = n-3$

$T(n-3) = T(n-3-3) + (n-3)^2$ — (3)

$T(n) = T(n-6) + (n-3)^2 + n^2$ — (4)

put $n = n-6$

$T(n-6) = T(n-3-6) + (n-6)^2$ — (5)

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

Date. ______
Page No. ______

Q6

$$T(n) = T(n-3k) + (n-3[k-1])^2 + (n-3[k-2])^2 + \cdots n^2$$

let $n-3k=1$

$$\frac{n-1}{3} = k$$

$$T(n) = T(1) + \left(n-3\left(\frac{n-1}{3}-1\right)\right)^2 + \left(n-3\left(\frac{n-1}{3}-2\right)\right)^2 \cdots n^2$$

$$T(n) = T(1) + (n-[(n-1)-3]^2) + (n-[n-1-6])^2 + (n-[n-1-9])^2 + \cdots n^2$$

$$T(n) = 1 + [3+1]^2 + [6+1]^2 \cdots n^2$$

$$T(n) = 1 + 4^2 + 7^2 \cdots n^2$$

$$T(n) = n^2 \cdots + 1$$

$$\underline{T(n) = \theta(n^2)}$$