Name: Aishwarya Kanbaj
Section: SPL16
UN. Roll: 201745
Class Roll no.: 6
Date.
Page No.

DAA Tutorial 1

Q1. Asymptotic notations are the mathematical notations used to describe running time of an algorithm when the input tends towards a particular value or a limiting value. Asymptotic notations are mainly categorised into following 3 types.

1) Big O notation → It gives the worst case complexity.
2) Omega notation → It gives the best case complexity.
3) Theta notation → It gives the average case complexity.

Eg. Bubble sort algorithm has $O(n)$ time complexity in best case & $O(n^2)$ time complexity in worst case & $O(n^2)$ in avg case.

Q2.
```
for (i = 1 to n)
{
    i = i * 2;
}
```

i = 1, 2, 4, 8 ..... n → G.P.

Date.
Page No.

$a_k = ar^{k-1}$, $a=1, r=2$

$a_k = 1 \cdot 2^{k-1}$

$n = 2^{k-1}$

$\log_2 n = k-1$

$k = 1 + \log_2 n$

$\therefore T(n) = O(\log_2 n + 1) = O(\log n)$ ans-

Q3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

$T(0) = 1$

$T(n) = 3T(n-1)$ — ①

put $n = n-1$ in eq ①

$T(n-1) = 3T(n-2)$ — ③

put ③ in ①

$T(n) = 3(3T(n-2)) = 3^2 T(n-2)$ — ④

put $n = n-2$ in eq ①

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

$T(n) = 3^2 \cdot 3T(n-3) = 3^3 T(n-3)$

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

let $n-k = 0$

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

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

Date.
Page No.

Q4. $T(n) = \{2T(n-1)-1$ if $n>0$, otherwise 1}

$T(n) = 2T(n-1)-1$ — (1)

$T(0) = 1$ — (2)

put $n = n-1$

$T(n) = 2T(n-2)-1$ — (3)

put (3) in (1)

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

$\Rightarrow 4T(n-2)-2-1 = 2^2T(n-2)-2-1$ — (4)

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

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

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

$\Rightarrow 2^3T(n-3)-2^2-2^1-1$

$T(n) = 2^kT(n-k)-2^{k-1}-2^{k-2}-2^{k-3}\ldots 2^0$

let $n-k=0$

$T(n) = 2^nT(n-n)-2^{n-1}-2^{n-2}-2^{n-3}\ldots 2^0$

$T(n) = 2^nT(0)-2^{n-1}-2^{n-2}-2^{n-3}\ldots 2^0$

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

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

$\{\because 2^{n-1}+2^{n-2}+\ldots+2^0 = 2^n-1\}$

$T(n) = 1$

$T(n) = O(1)$

Q5.
```
int i = 1, s = 1;
while (s <= n)
{
    i++;
    s = s + i;
    printf("#");
}
```

i = 1	s = 1	
i = 2	s = 3	s = 1+2
i = 3	s = 6	s = 1+2+3
i = 4	s = 10	s = 1+2+3+4

$s = 1+2+3+4+\cdots+k = \frac{k(k+1)}{2} > n \quad (\because s \le n)$

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

$k > \sqrt{n}$

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

Q6.
```
void function(int n)
{   int i, count = 0;
    for(i = 1; i*i <= n; i++)
    {
        count++;
    }
}
```

i = 1, 2, 3, . . . n

Date.
Page No.

$i^2 = 1, 2^2, 3^2 \dots n^2$

$i^2 = n$

$i = \sqrt{n}$

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

$a = 1, d = 1$

$a_k <= \sqrt{n}$

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

$\sqrt{n} = k$

$\therefore T(n) = O(\sqrt{n})$ Ans —

07.

```
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 n	$(\log n)^2$
n+1	$\log_2 n$	$(\log_2 n)^2$
2	⋮	⋮
⋮	log n	$(i \log_2 n)^2$
n		

$\frac{n}{2}$ + 1 times

$$O(i * k) = O\left(\frac{n}{2} + 1\right) * (\log n)^2$$

$$T(n) = O(n (\log n)^2) \text{ ans} -$$

Q8. function (int n)
{
if (n == 1)
return;
for (j = 1 to n)
{ for (j = 1 to n)
printf ("*");
}
function (n-3);
}

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

put n = n-3 in eq (1)

Date.
Page No.

$T(n-3) = T(n-6) + (n-3)^2$ — (2)

put $T(n-3)$ in (1)

$T(n) = T(n-6) + (n-3)^2 + n^2$ — (3)

put $n = n-6$ in (1)

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

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

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

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

put $n-3k = 1$

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

$T(n) = T(1) + n^2 + (n-3)^2 + (n-6)^2 + \dots (n-n+1)^2$

$T(n) = 1 + n^2 + (n-3)^2 + (n-6)^2 + \dots 1^2$

$T(n) = 6n^2 + k$

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

Q9.

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

i=1, j = 1, 2, 3, 4 ..... n times
i=2, j = 1, 3, 5, 7 ..... n/2 times
i=3, j = 1, 4, 7, 11 ..... n/3 times.
⋮
i=n j=1 ... 1 time.

$$\sum_{j=1}^{n} n + \frac{n}{2} + \frac{n}{3} + \cdots + 1$$

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

$= n(\log n)$

$T(n) = (n \log n)$

$T(n) = O(n \log n)$ as –

d10 $n^k = O(c^n)$

as $n^k <= 2c^n \quad \forall n > n_0$

for $n_0 = 1$

$c = 2$

$1^k <= a_2$

$n_0 = 1, c = 2.$