# DAA - 1

(1) Asymptotic notations are the mathematical notations used to describe the running time of an ~~alogorithm~~ algorithm when the input tends towards a particular value or a limiting value.

Eg - In bubble sort, when the input array is already sorted, the time taken by algorithm is linear i.e the best case ($\Omega$- notation)
omega- notation

But when the input array is in reverse condition, the algorithm takes the maximum time to sort the elements. i.e the worst case. (Big - O notation / O - notation).

When the input array is neither sorted nor in reverse order, then it takes ~~aveage~~ average time. ($\theta$ - notation)
Theta - notation

(2) $\sum_{i=1 (i=i*2)}^{n}$ 1 + 1 + ..... k times

$\therefore 2^k >= n$

$2^k = n$

taking log both side

$k \log 2 = \log n$

$k = \frac{\log n}{\log 2}$ $\left[ \log_b(x) = \frac{\log_a(x)}{\log_a(b)} \right]$

$k = \log_2 n$

Ans. $O(\log n)$

(3) $T(n) = \begin{cases} 3T(n-1) & n > 0 \\ 1 & n = 0 \end{cases}$

$T(n) = 3T(n-1)$ — (1)

Let $n = n-1$

putting $n$ in eq (1)

$T(n-1) = 3T(n-2)$ — (2)

putting (2) in (1)

$T(n) = 3^2 T(n-2)$ — (3)

Let $n = n-2$

putting $n$ in eq (1)

$T(n) = 3T(n-2)$ — (4)

putting (4) in (3)

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

~~Let $n-3=0$~~

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

Let $n-k=0$

$n = k$

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

$= O(3^n)$

(4) $T(n) = \begin{cases} 2T(n-1) - 1 & n > 0 \\ 1 & n = 0 \end{cases}$

$T(n) = 2T(n-1) - 1$ — (1)

~~put~~ Let $n = n-1$ in eq (1)

$T(n-1) = \cancel{2T} \; 2T(n-2) - 1$ — (2)

put this value in eq (1)

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

$T(n) = 4T(n-2) - 2 - 1$ — (3)

Let $n = n-2$

$T(n-2) = 2T(n-3) - 1$ — (4)

put this value in eq (3)

$T(n) = 4[2T(n-3) - 1] - 2 - 1$

$T(n) = 8T(n-3) - 4 - 2 - 1$

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

put $n - k = 0$

$n = k$

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

$= 2^n - [2^{n-1} + 2^{n-2} + \dots 2^0]$

$= 2^n - 2^{n-1}(1 - (\frac{1}{2})^n)\, 2$

$$= 2^n\left(1-\left(1-\left(\frac{1}{2}\right)^n\right)\right)$$

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

$$= 2^n\left(\frac{1}{2}\right)^n = 1$$

$$= O(1)$$

(5) $i = 1, 2, 3 \ldots$

$s = 1, 3, 6, 10 \ldots\ldots n$ — (1)

also $s = 1, 3, 6, 10 \ldots\ldots n$ — (2)

subtr. (1) - (2)

$0 = 1 + 2 + 3 + \ldots\ldots n - T_n$

$T_n = 1 + 2 + 3 + \ldots\ldots K$

for $K$ iterations

$1 + 2 + 3 \ldots\ldots + K <= n$

$$\frac{K(K+1)}{2} <= n$$

$$\frac{K^2 + K}{2} <= n$$

$$O(K^2) <= n$$

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

(6) $i^2 <= n$

$i <= \sqrt{n}$

$i = 1, 2, 3 \dots \sqrt{n}$

$$\sum_{i=1}^{n} 1 + 2 + 3 + \dots \sqrt{n}$$

$$T(n) = \frac{\sqrt{n} \times (\sqrt{n} + 1)}{2}$$

$$T(n) = \cancel{\frac{n \times \sqrt{n}}{2}} \quad \frac{n + \sqrt{n}}{2}$$

$$= O(n)$$

(7) for $k = k * 2$

$k = 1, 2, 4, 8 \dots n$

GP, $a = 1$ $r = 2$

$$= \frac{a(r^n - 1)}{r - 1}$$

$$= \frac{1(2^k - 1)}{1}$$

$n = 2^k$

$\Rightarrow \log n = k$

| i | j | k |
|---|---|---|
| 1 | $\log n$ | $\log n * \log n$ |
| 2 | $\log n$ | $\vdots$ |
| $\vdots$ | $\vdots$ | $\vdots$ |
| n | $\log n$ | $\log n * \log n$ |

$\Rightarrow \quad O(n * \log n * \log n)$

$O(n \log^2 n)$

(8) $T(n) = T(n/3) + n^2$

$a = 1, \; b = 3 \qquad f(n) = n^2$

$c = \log_3 1 = 0$

$\Rightarrow \; n^0 = 1 > [f(n) = n^2]$

$T(n) = \Theta(n^2)$

(9) for $i = 1 \Rightarrow j = 1, 2, 3, 4 \ldots \quad n = n$

for $i = 2 \Rightarrow j = 1, 3, 5 \ldots \quad n = n/2$

⋮

for $(i = n) \Rightarrow j = 1 \ldots$

$\Rightarrow \sum_{j=n}^{1} n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} \ldots + 1$

$\Rightarrow \sum_{j=n}^{1} n \left[1 + \frac{1}{2} + \frac{1}{3} + \ldots \frac{1}{n}\right]$

$\Rightarrow \sum_{j=n}^{1} n \log n$

$O(n \log n)$

(10) as given $n^k$ & $c^n$

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

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

as $n^k \leq a c^n$

$\forall n \geq n_0$ & some constant $a > 0$

for $n_0 = 1$
$c = 2$

$\Rightarrow 1^k \leq a 2^1$

$n_0 = 1$ & $c = 2$