Rahul Pandey
CST core
43

Date.
Page No.

① Asymptotic notation are the mathematical notations used to describe the running time of an 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 best case ($\Omega$ notation)(omega)

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

when input array is neither sorted nor in reverse order then it takes average time ($\Theta$-notation) theta notation

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

$\therefore\ 2^k = n$

taking log both side

$k \log 2 = \log n$

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

$k = \dfrac{\log n}{\log 2}$

$k = \log_2 n$

$O(\log n)$

(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)

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

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

let $n-k = 0$

$n = k$

$T'(n) = 3^n T(0)$

$= O(3^n)$

(4) $T(n) = 2T(n-1) - 1$ — (1)

put $n-1$ in eq (1)

$T(n-1) = 2T(n-2) - 1$ — (2)

put this in eq (1)

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

$T(n) = 4T(n-2) - 2 - 1$ — (3)

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

$T(n-1) = 2T(n-3)$ — (4)

put this in eq (4) (3)

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

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

Date.________
Page No.________

Generalized form

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

put $n-k=0$, $n=k$

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

$\therefore T(0) = 1$

$= 2^n - 2^{n-1} - 2^{n-2}$

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

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

$= 2^n - 2^{n-1} (1-(1/2)^n)2$

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

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

$T - O(1)$

(5) $i = 1, 2, 3, 4, 5, 6 \cdots$

sum of $S = 1 + 3 + 6 + 10 + 15 + 21$

~~also $1+3+6+10+ \cdots$~~ $T_{n-1} + T_n$ — (2)

~~$0 = 1+2+3+4+ \cdots + n - T_n$~~

$T_k = 1+2+3+4+\cdots k$

$T_k = \frac{1}{2}(k)(k+1)$

for k iterations $1+2+3+\cdots+k <= n$

$= \frac{k(k+1)}{2} <= n$, $\frac{k^2+k}{2} <= n$

$O(k^2) <= n$

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

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

Date.
Page No.

(6) $i^2 = n$

$i = \sqrt{n}$

$i = 1, 2, 3, 4, \ldots \sqrt{n}$

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

$= T(n) = O(n)$

(7) for $k = k * 2$

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

$GP = 1, 2, 4, 8, \ldots n$

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

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

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

$n = 2^{k-1}$ $\left\{ \begin{array}{l} \text{taking log on both sides } \log_2 n \\ k\log_2 2 - \log_2 1 \\ \log_2 n = k \end{array} \right.$

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

$\Rightarrow O(\log n * \log n) \Rightarrow O(n \log^2 n)$

Date.
Page No.

(8) function (int n)
{ if (n == 1)
return;    // o(1)
for (i = 1 to n)    // i = 1, 2, 3, 4 ... n → o(n)
{
for (j = 1 to n)
{
pf(" * ");
}
~~for~~ function (n-3);    // T(n/3)
}

Using master's method

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

$a = 1$    $b = 3$    $f(n) = n^2$

$c = \log_3 1 = 0$

$n^c = 1 > f(n)$

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

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

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

$i = 3 \Rightarrow j = 1, 4, 7 \ldots\ldots n = n/3$

for $i = n = j = 1$

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

$\sum_{j=n}^{1} n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}\right]$

$\sum_{j=n}^{1} n(\log n)$

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

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

(16) as given $n^k$ and $c^n$
relation b/w $n^k$ and $c^n$ is
$n^k = O(c^n)$

as $n^k \leq a c^n$
$\forall n \geq n_0$ some constant $a > 0$

for $n_0 = 1$
$c = 2$

~~$n_0 = 1$ and $c = 2$~~

$1^k \leq a2$
$n_0 = 1$ and $c = 2$