Ashish Dhek
CST
29

Amit
PAGE
DATE

DAA

(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. the best case (Ω - notation)
omega notation

But when the input array is in reverse condition, the algorithm takes the maximum time to sort the elements
i.e. word case (Big-O notation / O - notation)
when the input array is neither sorted nor in reverse order then it takes average time (Θ - notation)
Thela notation

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

$\therefore 2^k \geq = n$

$2^k = n$

taking log both side

$k \log 2 = \log n$

$k \log 2 = \log n$

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

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

$k = \log_2 n$

→ $O(\log n)$

PAGE
DATE

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

Let $n = n-1$

$T(n-1) = 3T(n-2)$ — (2)

$T(n) = 3[3T(n-2)]$ —

$T(n) = 3^2 T(n-2)$ — (4)

Let $n = n-2$

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

(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 $n-k = 0$

$n = k$

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

$= O(3^n)$

PAGE
DATE

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

Sum of $S = 1 + 3 + 6 + 10 + 15 + 21 \ldots n$ — (1)

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

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

$T_K = 1 + 2 + 3 + 4 + \ldots K$

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

for k iterations

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

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

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

$O(K^2) <= n$

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

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

(4) $T(n) = 2T(n-1) - 1$ — (1)

put $n-1$ in eq (1)

$T(n-1) = 2T(n-2) - 1$ — (2)

put this value in eq (1)

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

$T(n) = 4T(n-2) - 2 - 1$ — (3)

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

$T(n-2) = 2T(n-3) - 1$ — (4)

put this value in eq (3)

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

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

Sunil

PAGE
DATE

Generalized form

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

put $n-k=0 \Rightarrow n=k$

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

$\therefore T(0) = 1$

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

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

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

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

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

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

Time complexity $O(1)$

(6)

$i^2 = n$

$i = \sqrt{n}$

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

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

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

(7)

for $k = k * 2$

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

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

Sum of n-terms $= \dfrac{a(r^n-1)}{r-1}$

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

$n = 2^k - 1$

taking log on both sides.

$\log_2 n = k \log_2 2 - \log_2 1$

$\log_2 n = k$

i	j	k
1	$\log n$	$\log_n * \log n$
2	$\log n$	
:	:	
n	$\log n$	$\log_n * \log n$

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

(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)
         {
               printf (" * ");
         }
     function (n-3) ;             // T(n/3)
}
```

Using Master's Method

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

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

$c = \log_3 1 = 0$

$\Rightarrow n^c = 1 < f(n)$

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

Sunil

PAGE

DATE

(9) ~~void function~~

for $i=1 \Rightarrow j = 1, 2, 3, 4, \ldots n = n$.

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

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

for $i=n \Rightarrow j=1$

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

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

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

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

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

(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 $n_0 = 1$

$c = 2$

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

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