Rishav Kumar
G/10
2016959

DAA (TCS-409)

Tutorial - 1

1) Asymptotic notation are mathematical notations used to describe the running time of an algorithm where the input tends towards a particular value or a limiting value.
For 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)
But when the input array is in reverse condition the algorithm takes the maximum time to sort the elements i.e. worst case (O-notation/Big O notation)

2) $\sum_{i=1 (i=i\times2)}^{n}$ 1 + 1 + 1 + --- k times

$\therefore 2^k >= n$

$2^k = n$

taking log both sides

$k\log_2 2 = \log_2 n$

$k = \log_2 n$

$O(\log_2 n)$

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

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

putting (ii) in (i)

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

let $n = n-2$

~~Taking~~ putting n in eq (iii)

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

Put eq. (iv) in (iii)

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

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

let $n - K = 0$

$n = K$

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

$= 3^n \cdot 1$

$= 3^n$

$O(3^n)$

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

Sum of 3 $= 1 + 3 + 6 + 10 + 15 + \ldots \ldots n$ - (i)

also $1 + 3 + 6 + 10 + \ldots \ldots T_{n-1} + T_2$ - (ii)

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

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

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

For K iterations

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

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

$\frac{k^2+k}{2} \leq = n$

$k^2 = n$

$k = \sqrt{n}$

$O(\sqrt{n})$

4) $T(n) = 2T(n-1) - 1$ — (i)

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

$T(n-1) = 2T(n-2) - 1$ — (ii)

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

$= 4T(n-2) - 2 - 1$ — (iii)

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

$T(n-2) = 2T(n-3) - 1$ — (iv)

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

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

Generalised for

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

$n - k = 0$

$\Rightarrow n = k$

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

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

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

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

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

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

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 $k = k * 2$

$$k = 1, 2, 4, 8, \ldots n$$
$$n = \frac{a.(r^k - 1)}{r-1}$$
$$n = 1.\frac{(2^k - 1)}{(2-1)}$$
$$n = 2^k - 1$$
$$\log_2 n = k \log_2 2 - \log_2 1$$
$$k = \log_2 n$$

i	j	k
1	logn	logn * logn
2	logn	⋮
3	⋮	⋮
⋮	⋮	⋮
n	logn	logn * logn

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

8) function (int n)

```
{ if (n == 1)        // O(1)
     return;
  for (i = 1 to n)       // O(n)
  {
     for (j = 1 to n)    // O(n)
     {
        printf(" * ");
     }
  function(n-3);     T(n/3)
  }
```

using Master's Theorm

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

$a = 1, b = 3$

$c = \log_3 1 = 0$

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

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

9) 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, J = 1

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

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

$= n \log n$

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

10) as given n^k and c^n
relation b/w n^k and c^n is

$n^k = O(c^n)$ as $n^k \leq a.c^n$

$\forall\ n \geq n_0$.

for $n_0 = 1$

$c = 2$

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

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