Date: / /

DAA Assignment - tutorial - 1

Name - Satvik Mittal
Section - CST - SPL - 2
Semester - 4th
Roll no. - 34
University roll no. - 2017548

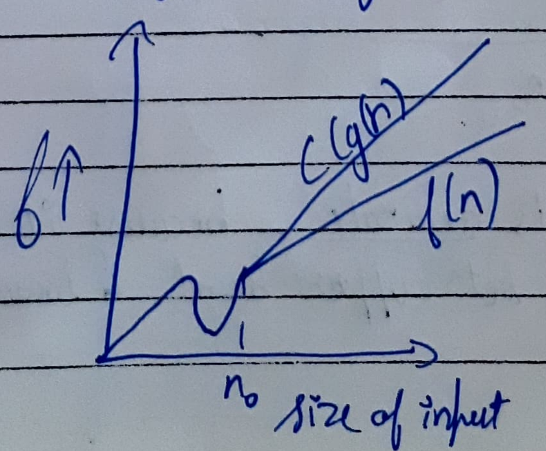
Satvik

Sol. 1. Asymptotic notations :-

- Help us to find the complexity of an algorithm when input is large.

- Big Oh - A function is said to be $O(g(n))$ iff there exists constant c and a constant n_0 such that

$$0 \le f(n) \le c\,g(n)$$

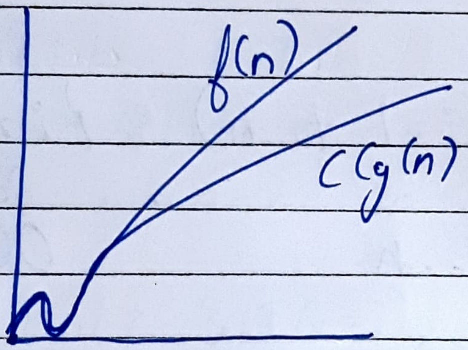


Date: / /

→ Big Omega (Ω).

• A function is said to $\Omega(g(n))$ iff there exists a constant C and a constant n_0 such that :-

$$0 \le C(g(n)) \le f(n)$$



→ Big theta (θ)

• A function is said to $\theta(g(n))$ iff there exists a constant C_1 and constant C_2 such that :-

$$C_2\, g(n) \le f(n) \le C_1\, g(n)$$

→ Small oh (o)

- $f(n) = O(g(n))$ iff

$$C g(n) > f(n)$$

Date: / /

- Small omega (ω)

- $f(n) = \omega g(n)$ iff :-

g(n) is lower bound of f(n)

$f(n) > c g(n)$

Q2.

Sol 2. for (i = 1 to n) { i = i + 2 }

$i = 1, 2, 4, 8 \ldots n$ $\qquad O(1)$

$\Rightarrow \sum_{i=1} 1 + 2 + 4 + 8 \ldots n$

GP k^{th} value $\rightarrow T_k = ar^{k-1}$

$= 1 \times 2^{k-1}$

$\Rightarrow n = 2^{k}$

$= 2n = 2^{k}$

$\log 2n = k \log 2$

$\log n + 1 = k$

$\Rightarrow O(k) = O(1 + \log n)$

$= O(\log n)$

Date: / /

Q3.
Sol3. $T(n) = 3T(n-1)$ if $n>0$, otherwise 1

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

Put $n = n-1$

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

from (1) and (2)

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

$= 9T(n-2)$ - (3)

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

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

$T(n) = 27(T(n-3))$.

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

Putting $n-k = 0$

$n = k$

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

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

$= 3^n \times 1$

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

Q4. Sol4- $T(n) = 2T(n-1)-1$ if $n>0$, otherwise 1

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

let $n = n-1$

Sah

Date: / /

$T(n-1) = 2T(n-2) - 1$ — ②

from ① and ②

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

$= 4T(n-2) - 3$ — ③

let $n = n-2$

$T(n-2) = 2T(n-3) - 1.$ — ④

from ③ & ④

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

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

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

$G.P = 2^{k-1} + 2^{k-2} + 2^{k-3} + \dots 1$

$a = 2^{k-1}$

$r = 1/2$

$\Rightarrow \dfrac{a(1-r^n)}{1-r}$

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

$= 2^k - 1$

Sato

Date: / /

Let $n-k=0$

$n=k$

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

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

$T(n) = O(1)$

Q5. Sol. S.

$S = 1+3+6+10+15+21 --- n$

Sum of $S = 1+3+6+10 ++ --- T_n$ - (1)

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

$T_k = 1+2+3+4+ --- +n$

for k iterations.

$1+2+3+ --- +k <= n$

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

$O(k^2) <= n$

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

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

Q6. Sol 6.

$i^2 <= n$

$\Rightarrow i <= \sqrt{n}$

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

$\sum_{i=1} 1+2+3+4+ ---- +\sqrt{n}$

Date: / /

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

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

$T(n) = O(n)$

Q7. Sol 7.

for $k = k * 2$

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

G.P $\Rightarrow$ $a = 1$, $r = 2$

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

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

$n \rightarrow 2^k$

$\log n \rightarrow k$

$\Rightarrow$ i	j	k
1	$\log n$	$\log n * \log n$
2	$\log n$	$\log n * \log n$
⋮	⋮	⋮
n	$\log n$	$\log n * \log n$

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

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

Date: / /

Q8 Sol.

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

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

$c = \log_3 1 = 0$

$n^0 = 1 > f(n) = n^2)$

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

Q9. Sol

$=$ 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 $i = n \Rightarrow j = 1$

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

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

$\sum_{j=n} n[\log n]$

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

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

Date: / /

Q10.
Sol. Given n^k and c^n
relation b/w n^k & c^n is

$n^k = O(c^n)$
or $n^k \le ac^n$
$\forall\ n \ge n_0$ and some constant $a > 0$
for $n_0 = 1$
$c = 2$
$1^k \le a2^1$

$n_0 = 1$ & $c = 2$