Name → Dhruva Bisht , Roll No → 25

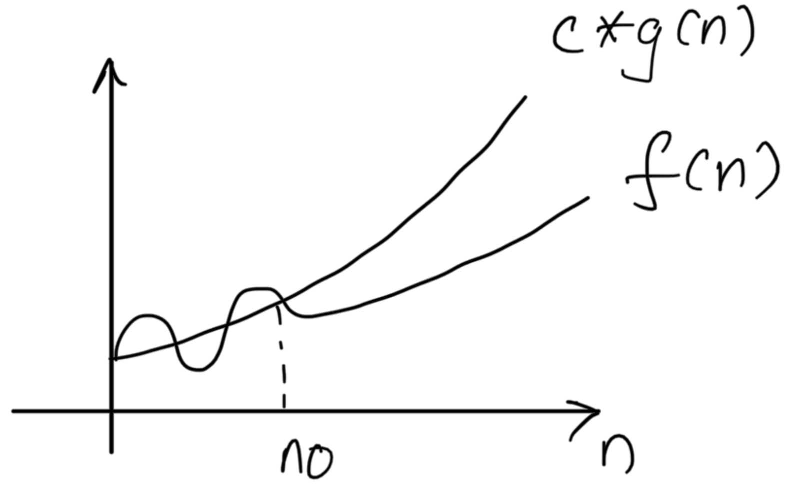
Section → DS

DAA Tutorial - 1

Q1) Asymptotic Notation → These are basically a language to express the required time & space used by an algorithm to solve a given problem.

① Big-O-Notation → It is a notation for the worst case analysis of an algorithm, (upper-bound). According to it $f(n) = O(g(n))$

if and only if n_0 & c are such that $0 \leq f(n) \leq c * g(n)$ for all $n \geq n_0$.



eg → $n + n^2 = O(n^2)$

here $f(n) = n + n^2$, $g(n) = n^2$

$n + n^2 \leq n^2 + n^2$ ($\because n < n^2, n^2 = n^2$)

$n + n^2 \leq 2n^2$ (here $c = 2$) for $n_0 = 1$

$\therefore f(n) = O(g(n)) \Rightarrow n + n^2 = O(n^2)$

② <u>Big Theta</u> (Θ) :→ For avg case,

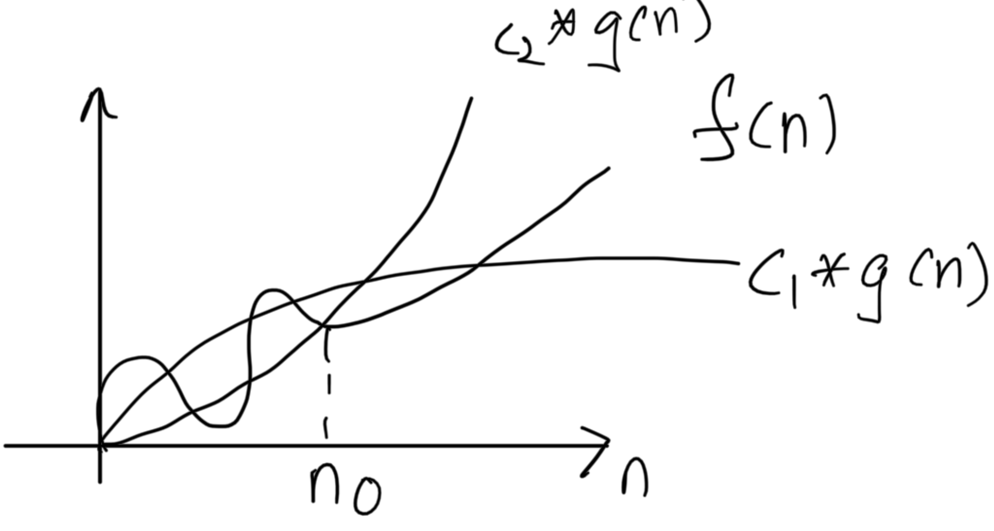
time complexity (tightly bound)
for any two functions $f(n)$ & $g(n)$

$f(n) = \Theta(g(n))$ if → there exists

n_0, c_1, c_2 as ∋

$$0 \leq c_1 * g(n) \leq f(n) \leq c_2 * g(n)$$

for $n \geq n_0$

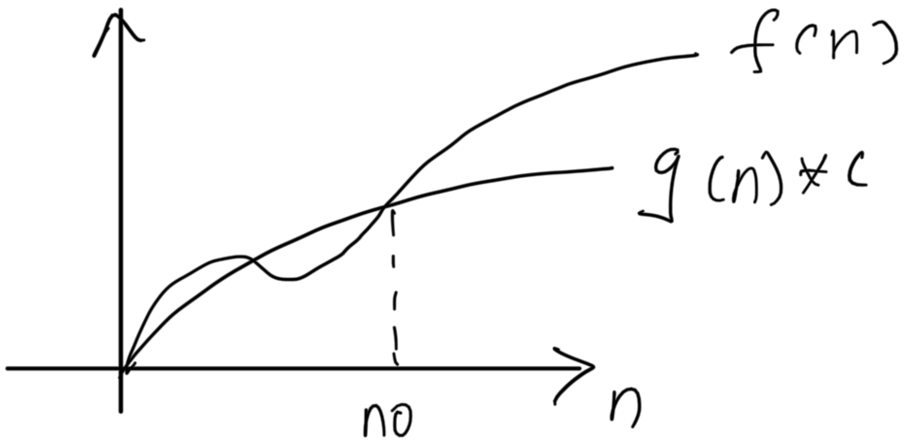


③ <u>Big Omega</u> (Ω) → for best

case complexity, (lower bound)

$f(n) = \Omega(g(n))$ if $\exists$ n_0, c_1

$\ni$ $0 \leq c_1 * g(n) \leq f(n)$ $\forall$ $n \geq n_0$



Q2) TC for (i=1 to n) { i = i*2 }

Series → 1, 2, 4, 8 .... n (G.P)

$a = 1, \; r = 2$

$t_k = a r^{k-1} \Rightarrow n = a * 2^{k-1}$

$n = 2^{k-1}$

$2^k = 2n$

$k = 2 \log_2 n \quad \therefore TC = O(\log_2 n)$

Q3) $T(n) = \{3T(n-1)) \text{ if } n > 0, \text{ else } 1\}$

$T(n) = 3T(n-1)$ --- ①

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

$T(n) = 3^2 T(n-2) \Rightarrow T(n) = 3^3 T(n-3)$

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

$T(n) = 3^n T(0) = 3^n \Rightarrow T(= O(3^n)$

Q4) $T(n) = \{2T(n-1)-1, \{n>0, 1\}$

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

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

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

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

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

$\Rightarrow 2^2(2T(n-3)-1)-2-1) = 2^3(T(n-3))$
-2^2-2-1

or) $T(n) = 2^k T(n-k) - 2^{k-k} - 2^{k-2} \ldots - 2^0$

$T(0) = 1$, let $n-k=0$, $k=n$

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

$= 2^n - 2^{n-1} - 2^{n-2} \dots\dots 2^0$ {G.P}

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

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

$\therefore O(1)$

Q5)
```
int i=1, s=1;
while (s<=n) {
    i++; s=s+i;
    printf("#");
}
```

Seriese = 1, 3, 6, 10, 15 ..... n

1st It ⇒ S = S + 1
2nd ⇒ S = S + 1 + 2   till n

$$\therefore \quad \frac{k*(k+1)}{2} <= n \Rightarrow k^2 <= n$$

$$\Rightarrow \quad k = O(\sqrt{n}) = TC$$

Q6) for (i=1 ; i * i <= n ; i++)
count ++

let loop run till $i = k$

$k^2 <= n$; $k <= \sqrt{n}$

$\Rightarrow$ T.C = $O(\sqrt{n})$

Q7) for (i = n/2 ; i <= n ; i++)
{
for (j=1 ; j <= n; j = j*2)
for (k=1 ; k <= n ; k = k*2)

TC = $O(n \log^2 n)$

Q8) Recurrance Relation $\Rightarrow T(n) = T(n-3) + n^2$

$$T(n) = T(n-6) + 2 + n^2$$
$$= T(n-9) + 3n^2$$

$$= T(n-3k) + kn^2$$

$T(1) = 0$; $n - 3k = 1$; $k = \frac{n-1}{3}$

$$T(n) = T(1) + \frac{(n-1)}{3} n^2$$

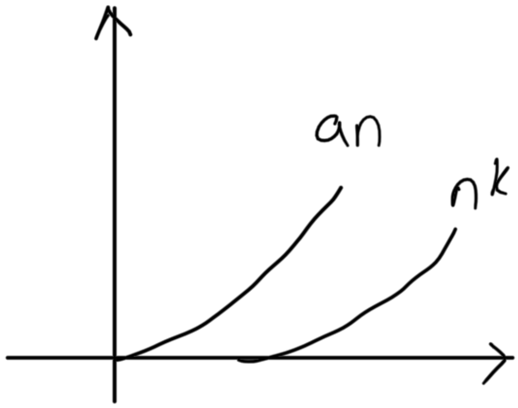
$\Rightarrow TC = O(n^3)$

Q9)

i	j	times
1	$1 \to n$	n
2	$1 \to n$	$n/3$
$\vdots$		
n	$1 \to n$	1
		$n \log n$

$\therefore TC = O(n \log n)$

Q10) $n^k = O(a^n)$



$n^k \le a^n, c \;\forall\; c > 0$

& $n \geq n_0$

let $n = n_0$

$n_0^k \le c \cdot 2^{n_0}$

[so let $k = a = 3$; $n_0^3 \le c \cdot 3^{n_0}$

so $c \geq 1$ & $n_0 \geq 1$]

Name → Dhruva Bisht , Roll No → 25

Section → DS

DAA Tutorial - 2

Q11) Seriese ⇒ 0, 1, 3, 6, 10, 15 . . .

at last iteration

$$n = 0 + 1 + 2 + 3 + 4 + 5 \ldots + k$$

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

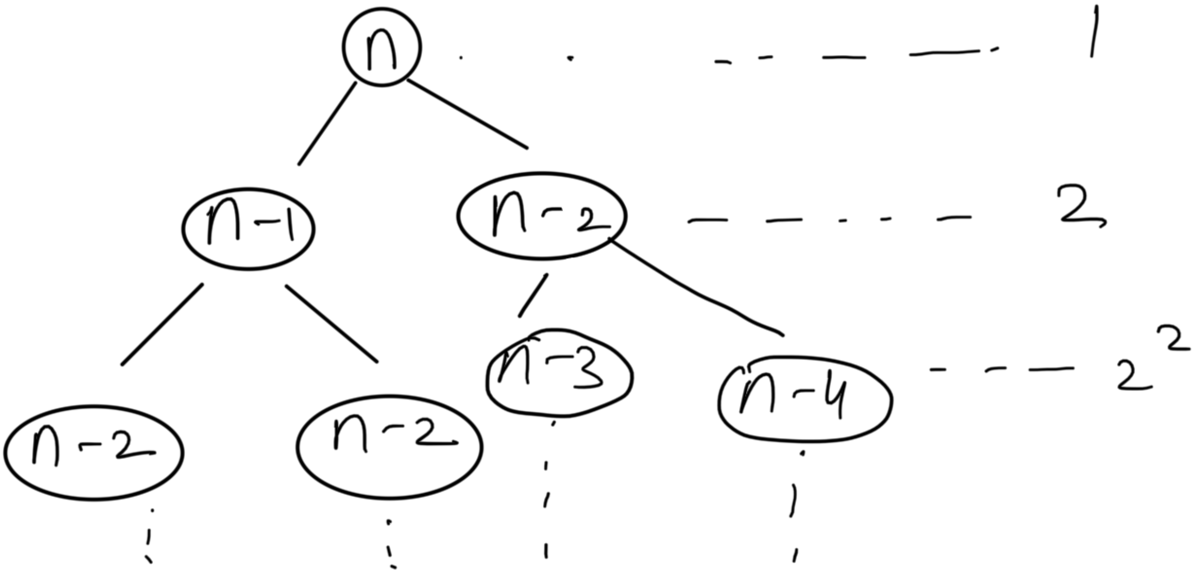
$$n = \frac{k^2+1}{2} \Rightarrow n \cong k^2$$

$$\Rightarrow k \cong \sqrt{n} \Rightarrow TC = O(\sqrt{n})$$

Q12) Recurrance Relation for

fibonacci ↓

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



$$TC = 1 + 2 + 4 + \dots + 2^n = 1\,\frac{(2^{n+1} - 2)}{2-1}$$

$= 2^{n+1} - 1$ so $TC = O(2^n)$

SC → is proportional to height of recurrence tree. ⇒ $O(n)$

Q13) (i) <u>n logn</u>

```
for (i to n)
{
    for (j=1, j<=n ; j *=2)

            O(1) statements

}
```

(ii) $\underline{n^3}$ → for (i to n)
for (j to n)
for (k to n)
O(1) statements)

(iii) log(log n) →

i = n
while (i > 0)
{
$i = \sqrt{i}$;
}

Q14) $T(n) = T(n/4) + T(n/2) + cn^2$

n — — — — cn^2

$n/4$ $\quad$ $n/2$ — — — $5cn^2/16$

$\frac{n}{16} \quad \frac{n}{8} \quad \frac{n}{8} \quad \frac{n}{4} \; ---- \; \frac{25cn^2}{256}$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

$$T(n) = c\left(n^2 + \frac{5n^2}{16} + \frac{25n^2}{256} + \dots\right)$$

$$r = \frac{5}{16}, \quad S_n = \frac{1}{1-r} \Rightarrow T(n) = cn^2\left(1 + \frac{5}{16} \dots\right)$$

$$\Rightarrow c(n^2)\left(\frac{16}{11}\right) \Rightarrow T.C = \Theta(n^2)$$

Q15)

i	j	times
1	$1 \to n$	$n-1$
2	$1 \to n$	$(n-1)/2$
3	$1 \to n$	$(n-1)/3$
$\vdots$	$\vdots$	$\vdots$
n	$1 \to n$	$n-1/n$
		$n \log n$

$TC = O(n \log n)$

Q16) $i = 2, 2^{k}, 2^{k^2} \dots 2^{k^x}$

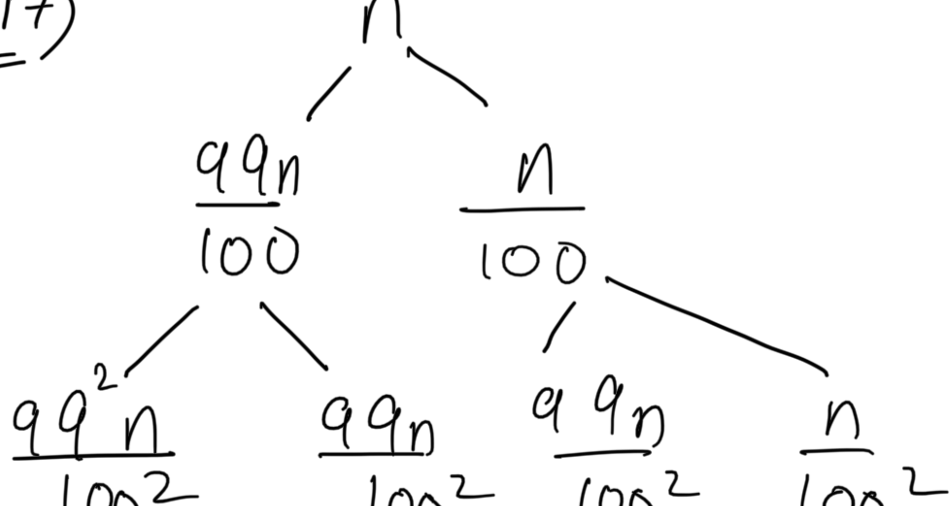
$$n = 2^{k^x}$$

$$\log n = k^x \log 2$$

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

$$x = \frac{\log \log n}{\log 2 \log k} \Rightarrow TC = O(\log \log n)$$

Q17)



$TC \Rightarrow \log_{\frac{100}{99}} n \simeq \log n$

$n = \left(\frac{99}{100}\right)^k \Rightarrow k = \log_{\frac{100}{99}} n$

$T(n) = n\left(\frac{\log_{\frac{100}{99}}}{100}\right)^n = O(n \log_{99} n)$

Q18) (Inc growth)

(a.) $100 < \log\log n < \log n < \sqrt{n} < n$

$< n\log n < n^2 < 2^n < 2^{2n} < 4^n < n!$

(b.) $1 < \log\log n < \sqrt{\log(n)} < \log n <$

$\log 2n < 2\log n < n < 2n < 4n < n^2 \log n$

$< \log(n) < 2^{2n} < n!$

(c.) $36 < \log_8 n < \log_2 n < 5n < n \log_8(n)$

$< n \log_2 n < 8n^2 < 7n^3 < \log n! <$

$8^{2n} < n!$