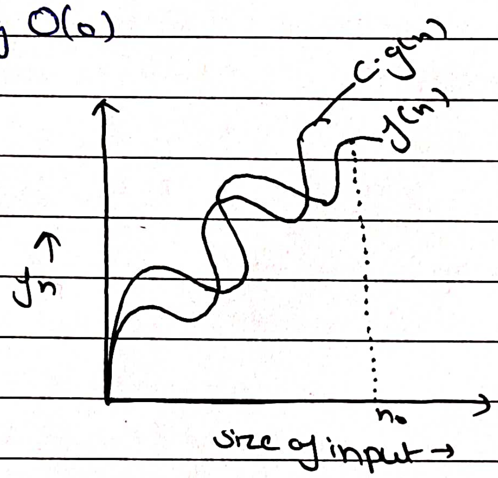
Q1) Asymptotic Notations

↳ Tending to Infinity

They help you find the complexity of an algorithm when input is very large.

① Big O(o)



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

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

$\forall\, n \ge n_0$

for some constant $c > 0$

→ $g(n)$ is tight upper bound of $f(n)$

② Big Omega(Ω)

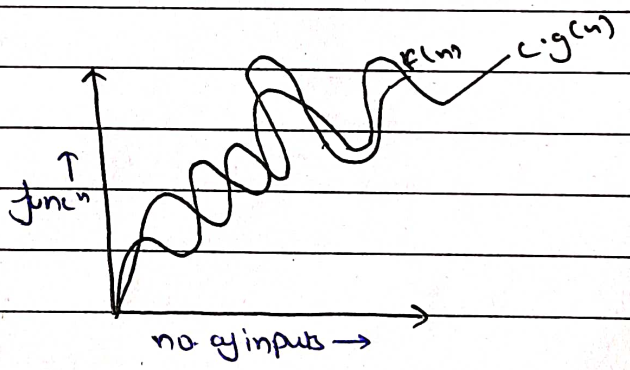
$f(n) = \Omega(g(n))$

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

$f(n) = \Omega(g(n))$

if $f(n) \ge c \cdot g(n)$

$\forall\, n \ge n_0$ for some constant $c > 0$

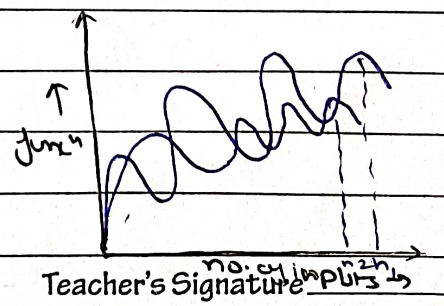


③ Theta(Θ)

$f(n) = \Theta(g(n))$

$g(n)$ is both 'tight' upper and lower bound of $f(n)$

$f(n) = \Theta(g(n))$



Date ____________ Page No. ____________

if $c_1 g(n) \le f(n) \le c_2 \cdot g(n)$

$\forall n \ge c\,max(n_1, n_2)$

for some constant $c_1 > 0$ and $c_2 > 0$

④ Small o(o)

$f(n) = o(g(n))$

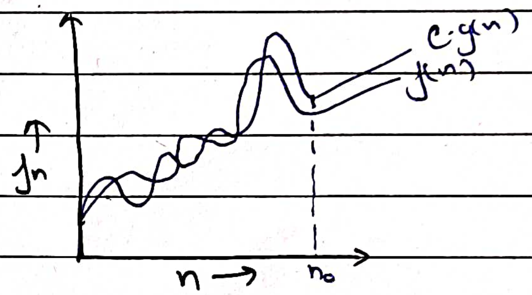
g(n) is upper bound of fn f(n).

$f(n) = o(g(n))$

when $f(n) < c \cdot g(n)$

$\forall n > n_0$

& $\forall c > 0$



⑤ Small omega(ω)

$f(n) = \omega(g(n))$

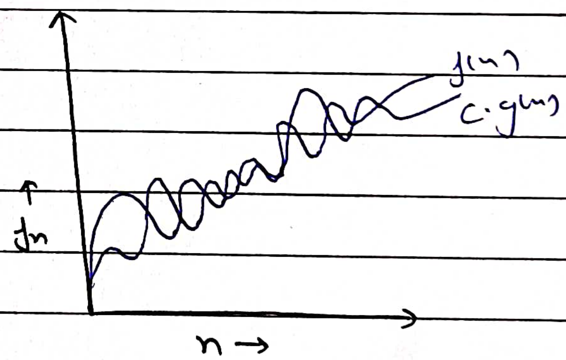
g(n) is lower bound of fn f(n)

$f(n) = \omega(g(n))$

when $f(n) > c\,g(n)$

$\forall n > n_0$

$\forall c > 0$



APPU Teacher's Signature ____________

Q2) What should be T.C of
for(i=1 to n) {i=i*2}

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

GP kth value $\Rightarrow T_k = ar^{k-1}$
$\Rightarrow 1 \times 2^{k-1}$
$n = 2^k$
$2n = 2^k$
$\log 2n = k \log 2$
$\Rightarrow \log 2 + \log n = k \log 2$
$\Rightarrow \log n + 1 = k$

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

Q3) $T(n) = \{3T(n-1)$ 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

$\Rightarrow T(n) = 3(3T(n-2))$
$= 9T(n-2)$ — ③

Date__________ Page No.__________

putting $n = n-2$ in eq. ①

$T(n-2) = 3(T(n-3))$ — ④

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

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

putting $n-k = 0$

$\Rightarrow n = k$

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

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

$T(n) = 3^n \times 1$ $[T(0) = 1]$

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

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

$T(n) = 2T(n-1) - 1$ — ①

let $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — ②

from ① & ②

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

$\Rightarrow T(n) = 4T(n-2) - 3$ — ③

let $n = n-2$

$\Rightarrow T(n-2) = 2T(n-3) - 1$ — ④

from ③ and ④

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

APPU Teacher's Signature__________

Date

Page No.

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

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

$GP = 2^{k-1} + 2^{k-2} + 2^{k-3} + \cdots + 1$

$a = 2^{k-1}$

$r = \frac{1}{2}$

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

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

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

$$= 2^k - 1$$

let $n-k=0$

$\Rightarrow n=k$

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

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

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

$\Rightarrow T(n) = O(1)$

Teacher's Signature

Date

Page No.

Q5) i = 1 2 3 4 5 6

$S = 1 + 3 + 6 + 10 + 15 + 21 + \dots + n$

sum of $S = 1 + 3 + 6 + 10 + \dots + n$ — ①

also $S = 1 + 3 + 6 + 10 + \dots + T_{n-1} + T_n$ — ②

from ① − ②

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

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

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

for k iterations

$1 + 2 + 3 \dots + k <= n$

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

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

$\Rightarrow O(k^2) <= n$

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

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

Q6) $O(1)$

as $i^2 <= n$

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

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

Date______ Page No.______

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

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

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

$$T(n) = O(n) \quad //$$

Q7) for $k = k^2$

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

$GP \Rightarrow a=1, \; r=2$

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

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

$n \Rightarrow 2^k$

$\log n \Rightarrow k$

$\Rightarrow$

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

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

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

APPU Teacher's Signature______

Date________ Page No.________

Q18| Time Complexity of

```
function(int n)
{ int(n==1)
    return;                    //O(1)
  for(i=1 to n)                //i=1, 2,3,4..... n => O(n)
  { for(j=1 to n)              //j=1,2,3,4.... n*n => O(n^2)
    { print('*');
    }
  }
  function(n-3);               T(n/3)
}
```

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

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

$c = \log_3 1 = 0$

$\Rightarrow n^0 = 1 \quad > (f(n) = n^2)$

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

Q19| Time complexity of

```
void function(int n)
{ for(i=1 to n)                //O(n)
  { for(j=1; j<=n; j=j+1)
      print("*")               //O(n)
  }
}
```

Teacher's Signature________

APPU

for $i=1 \Rightarrow j = 1, 2, 3, 4 \ldots\ldots n \quad = n$
for $i=2 \Rightarrow j = 1, 3, 5 \ldots\ldots n \quad = n/2$
for $i=3 \Rightarrow j = 1, 4, 7 \ldots\ldots n \quad = n/3$

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

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

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

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

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

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

Q1(b) for functions, n^k and c^n, what is the asymptotic relation between these functions?
assume that $k >= 1$ and $c > 1$ are constant.
Find out the value of c and n_0 for which relation holds

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 $n_0 = 1$
$c = 2$

$\Rightarrow 1^k \le a2^1$

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