(Name- Krishna Rathi F-62)

Tutorial- 1

Ques 1) What do you understand by Asymptotic notation, define different asymptotic notation with example.

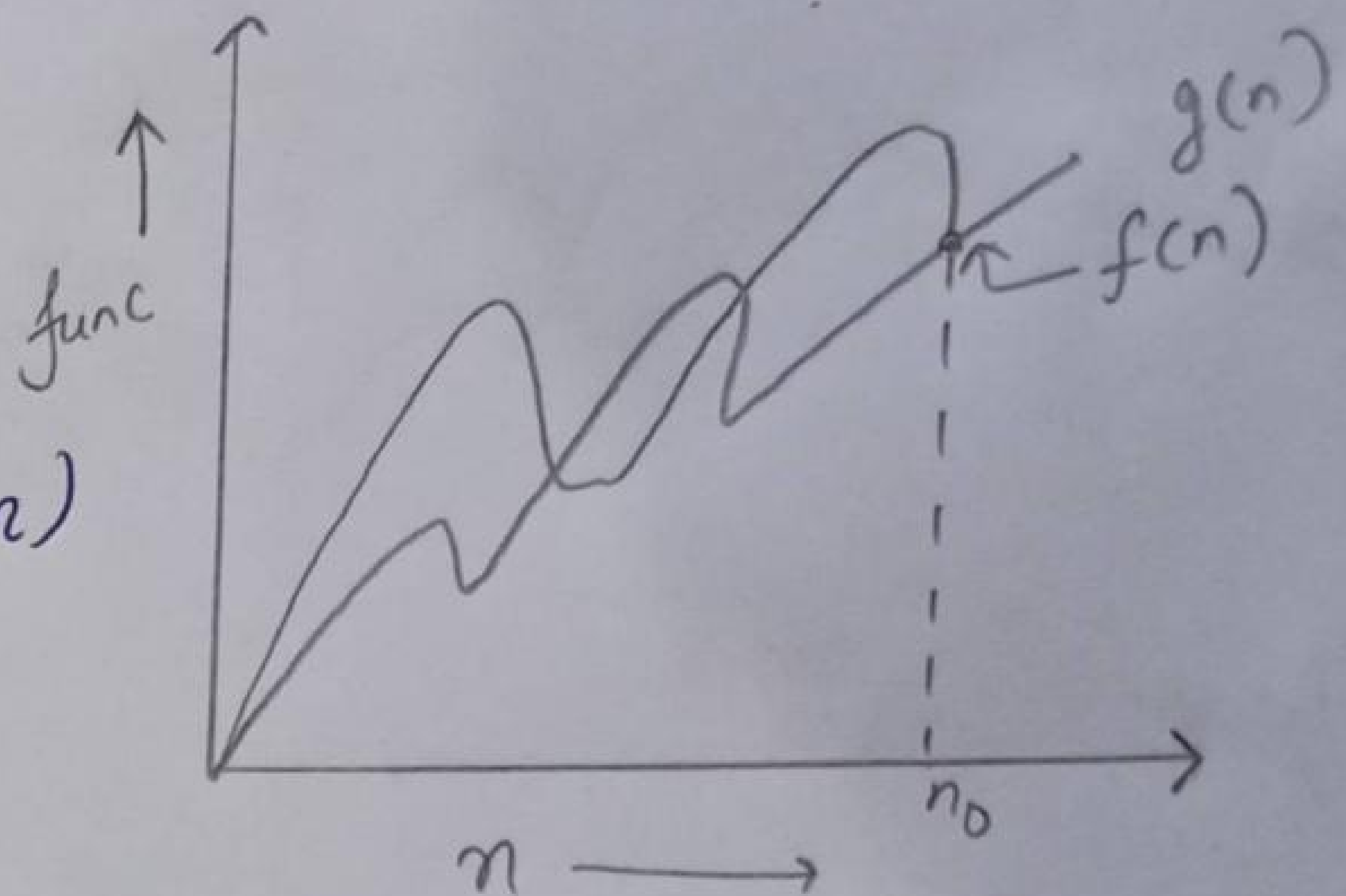
i) **Big O(n)**

$f(n) \Rightarrow O(g(n))$

if $f(n) \leq g(n) \times c \ \forall n > n_0$

for same constant, $c > 0$

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



Eg- $f(n) \Rightarrow n^2+n$

$g(n) \Rightarrow n^3$

$n^2+n \leq c * n^3$

$n^2+n = O(n^3)$

(ii) **Big Omega (Ω)**

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

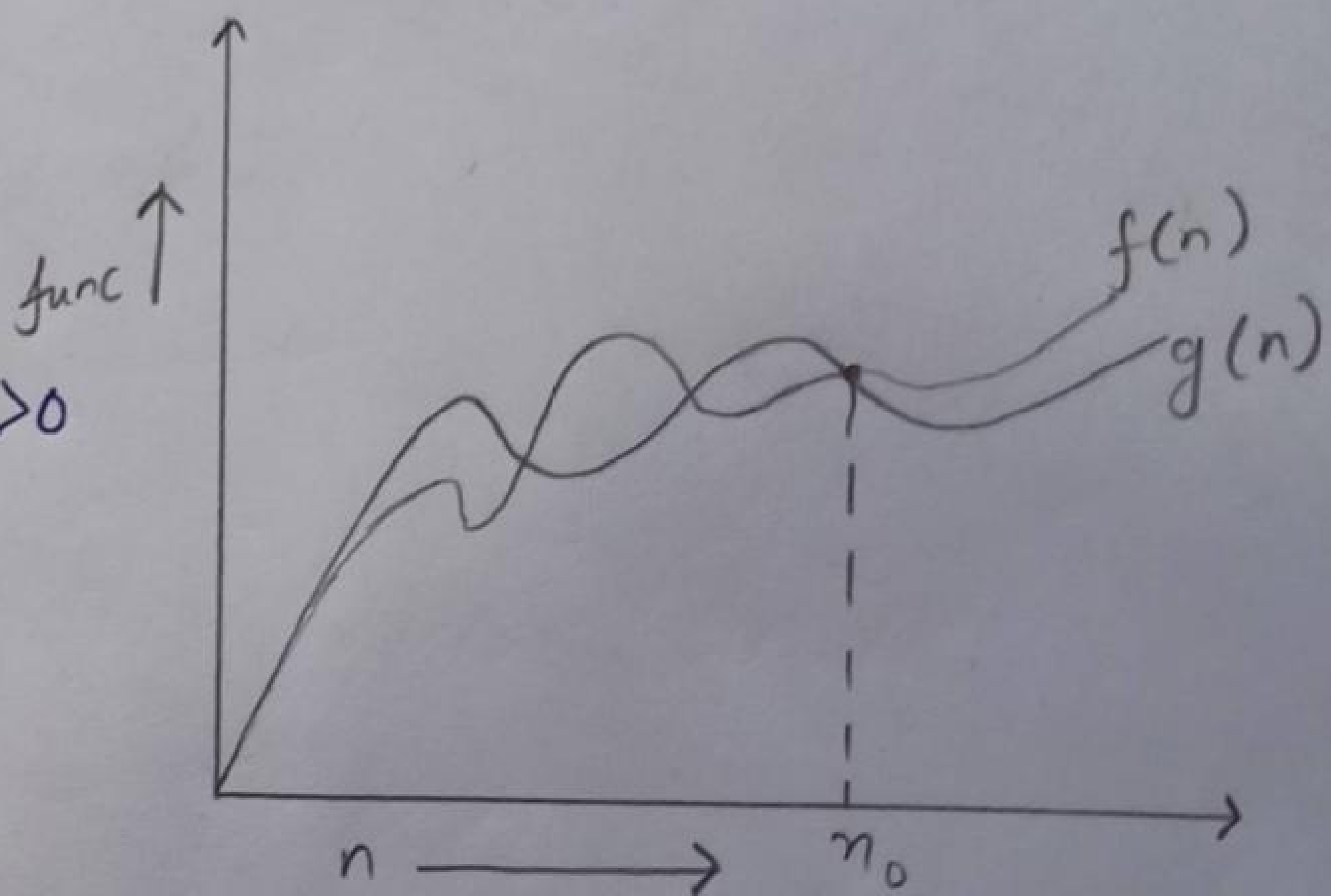
means $g(n)$ is "tight" lowerbound of $f(n)$ i.e $f(n)$ can go beyond $g(n)$

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

if and only if

$f(n) \geq c.g(n)$

$\forall n_2 > n_0$ and c = constant >0



ex:- $f(n) \Rightarrow n^3+4n^2$

$g(n) \Rightarrow n^2$

i.e $f(n) \geq c*g(n)$

$n^3+4n^2 = \Omega(n^2)$

(iii) **Big Thetha (θ)**

When $f(n) = \theta(g(n))$ gives the tight upperbound and lowerbound both,

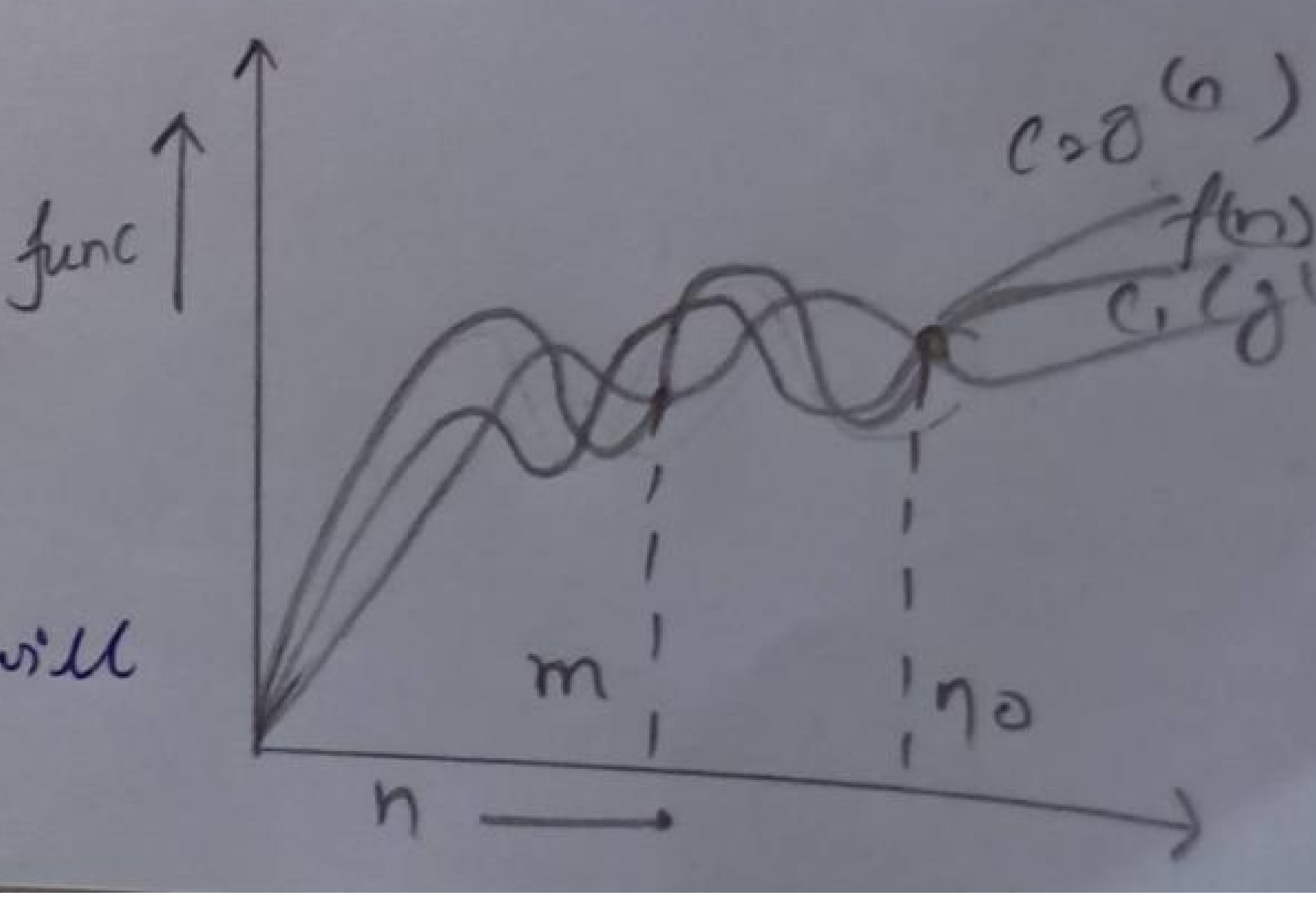
i.e $f(n) = \theta(g(n))$

if and only if

$c_1 * g(n_1) \leq f(n) \leq c_2 * g(n_2)$

for all $n \geq \max(n_1, n_2)$, some constant $c_1 > 0$ & $c_2 > 0$

i.e $f(n)$ can never go beyond $c_2 g(n)$ and will never come down of $c_1 g(n)$

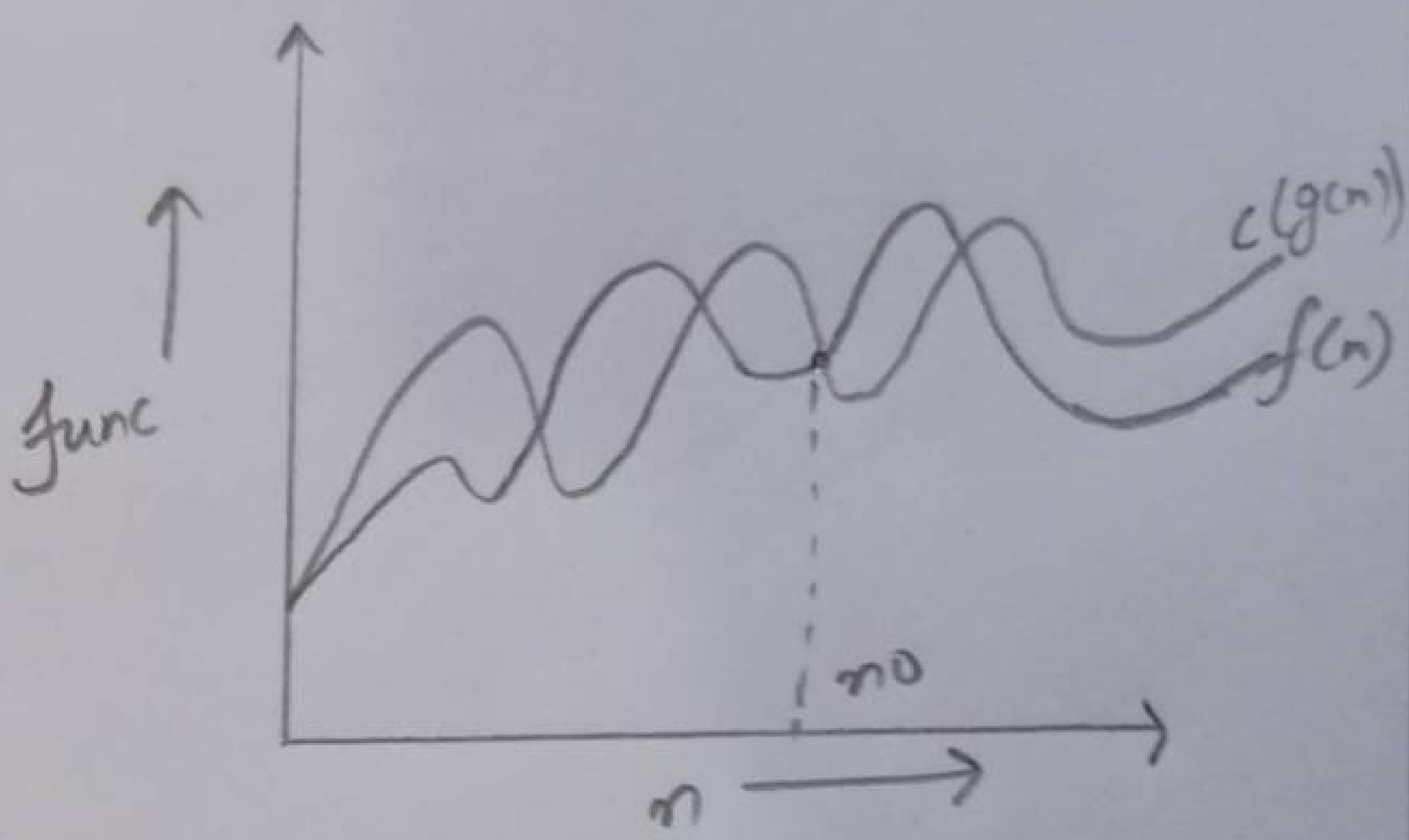


Scanned by TapScanner

$3n+2 = \theta(n)$ as $3n+2 \geq 3n$ & $3n+2 \leq 4n$ for n, $c_1 = 3, c_2 = 4$ & $n_0 = 2$

) **Small $O(\theta)$**

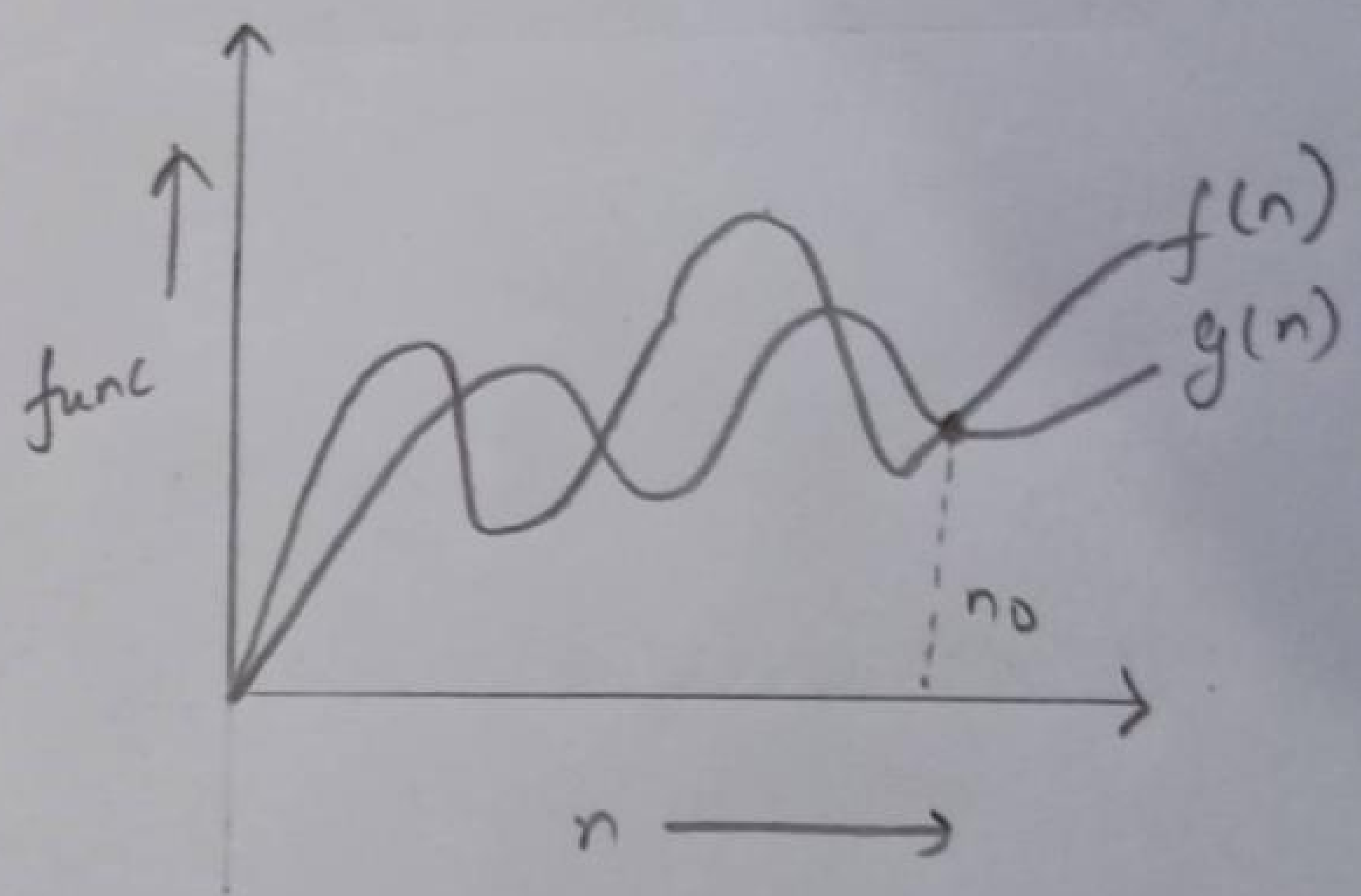
When $f(n) = O\,g(n)$ gives the upper bound
i.e $f(n) = O\,g(n)$
if and only if
$f(n) < c * g(n)$
$\forall\ n > n_0$ & $n > 0$



Ex: $f(n) = n^2$; $g(n) = n^3$
$f(n) < c * g(n)$
$n^2 = O(n^3)$

v) **Small Omega (ω)**

It gives the 'lower bound' i.e
$f(n) = \omega(g(n))$
where $g(n)$ is lower bound of $f(n)$
if and only if $f(n) > c * g(n)$
$\forall\ n > n_0$ & some constant, $c > 0$



Ques2) What should be time complexity of :

```
for (int i=1 to n)
{
    i = i*2 ;  → O(1)
}
```

Ans) for $i \Rightarrow 1, 2, 4, 6, 8 \ldots n$ times
i.e series is a GP
So, $a = 1$, $r = 2/1$
k^{th} value of GP ;
$t_k = a r^{k-1}$
$t_k = 1(2)^{k-1}$
$2n = 2^k$
$\log_2(2n) = k \log 2$
$\log_2 2 + \log_2 n = k$
$\log_2 n + 1 = k$ (Neglecting '1')

So, Time complexity $T(n) \Rightarrow \boxed{O(\log_2 n)}$

Ques 3) $T(n) = \{ 3T(n-1)$ if $n>0$
otherwise 1
$\}$

Ans) i.e $T(n) \Rightarrow 3T(n-1)$ —— (1)
$T(n) \Rightarrow 1$
put $n \Rightarrow n-1$ in (1)
$T(n-1) \Rightarrow 3T(n-2)$ —— (2)
put (2) in (1)
$T(n) \Rightarrow 3 \times 3T(n-2)$
$T(n) \Rightarrow 9T(n-2)$ —— (3)
put $n \Rightarrow n-2$ in (1)
$T(n-2) = 3T(n-3)$
put in (3)
$T(n) = 27T(n-3)$ —— (4)

Generalising series,
$T(k) = 3^k T(n-k)$ —— (5)
for k^{th} terms, Let $n-k=1$ (Base case)
$k = n-1$
put in (5)
$T(n) = 3^{n-1} T(1)$
$T(n) = 3^{n-1}$ (Neglecting 3')
$\boxed{T(n) = O(3^n)}$

Ques 4) $T(n) = \{ 2T(n-1) - 1$ if $n>0$,
otherwise 1
$\}$

Ans) $T(n) = 2T(n-1) - 1$ —— (1)
put $n = n-1$
$T(n-1) = 2T(n-2) - 1$ —— (2)
put in (1)
$T(n) = 2 \times (2T(n-2) - 1) - 1$
$= 4T(n-2) - 2 - 1$ —— (3)
put $n = n-2$ in (1)
$T(n-2) = 2T(n-3) - 1$
Put in (1)
$T(n) = 8T(n-3) - 4 - 2 - 1$ —— (4)
Generalising Series —

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

$\Rightarrow$ k^{th} term

Let $n-k=1$

$k = n-1$

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

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

i.e series in GP

$a = 1/2$, $r = 1/2$

So, $T(n) = 2^{n-1} \left(1 - \left(\frac{1/2\,(1-1/2)^{n-1}}{1-1/2}\right)\right)$

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

$= \frac{2^{n-1}}{2^{n-1}}$

$\boxed{T(n) = O(1)}$

Ques5) what should be time complexity of

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

Ans) $i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \dots$

$s = 1 + 3 + 6 + 10 + 15 + \dots$

Sum of $s = 1 + 3 + 6 + 10 + \dots + n$ —— (1)

Also $S = 1 + 3 + 6 + 10 + \dots T(n-1) + T_n$ —— (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

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

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

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

$O(k^2) <= n$

$K = O\sqrt{n}$

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

Scanned by TapScanner

Ques 6) Time Complexity of

```
void f(int n)
{
 int i , count=0;
for(i=1 ; i<=n ; ++1)
}
```

Ans) As $i^2 = n$

$i = \sqrt{n}$

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

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

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

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

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

Ques 7) Time complexity of

```
void f(int n)
{
  int i, j, k, count =0;
for (int i= n/2 ; i<=n ; +i)
 for (j=1 ; j<=n ; j= j*2 )
 for (k=1 ; k<=n ; k= k*2)
       count ++ ;
 }
```

Ans) Since , for $k = k^2$

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

$\because$ Series is in GP

So, $a=1$, $r=2$

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

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

$n = 2^k - 1$

$n+1 = 2^k$

$\log_2(n) = k$

i	j	k
1	$\log(n)$	$\log(n) * \log(n)$
2	$\log(n)$	$\log(n) * \log(n)$
⋮	⋮	⋮
n	$\log(n)$	$\log n * \log(n)$

$T.C \Rightarrow O(n * \log n * \log n)$

$\Rightarrow \boxed{O(n \log^2(n))}$ Ans

Ques 2) Time complexity of

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

Ans) for (i=1 to n)

we get $j = n$ times every turn

$\therefore \; i * j = n^2$

Now,

$T(n) = n^2 + T(n-3);$

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

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

and $T(1) = 1;$

Now, substitute each value in $T(n)$

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \dots + 1$

Let

$k^n - 3k = 1$

$k = (n-1)/3$ total times $= k+1$

$T(n) = n^2 + (n-3)^2 + (n-6)^2 + \dots + 1$

$T(n) \simeq kn^2$

$T(n) \simeq (k-1)/3 * n^2$

So, $\boxed{T(n) = O(n^3)}$ Ans

9) Time complexity of :-

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

Ans) for $i=1$ $\quad j = 1+2+ \dots \quad (n \geq j+i)$

$i=2 \quad j = 1+3+5+ \dots \quad (n \geq j+i)$

$i=3 \quad j = 1+4+7 \dots \quad (n \geq j+i)$

n^{th} term of AP is

$T(n) = a + d * m$

$T(m) = 1 + d \times m$

$(n-1)/d = n$

for $i=1 \quad (n-1)/1$ times

$i=2 \quad (n-1)/2$ times

$\vdots$

$i = n-1$

we get,

$T(n) = i_1 j_1 + i_2 j_2 + \dots i_{n-1} j_{n-1}$

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

$= n + n/2 + n/3 + \dots n/n-1$ ——— $n \times 1$

$= n[1 + 1/2 + 1/3 + \dots 1/n-1] - n * 1$

$= n \times \log n - n + 1$

Since

$\int 1/x = \log x$

$\boxed{T(n) = O(n \log n)}$

Ques 10) For the function n^{-1} R & c^n, what is the asymptotic relationship between these functions? Assume that $k > 1$ & $c > 1$ are constants. Find out the value of c & no. of which relationship holds.

Ans) As given n^k and c^n

Relationship b/w n^k & c^n is

$$n^k = O(c^n)$$

$$n^k \le a(c^n)$$

$\forall$ $n \geq n_0$ & constant, $a > 0$

for $n_0 = 1$; $c = 2$

$\Rightarrow 1^k < a^2$

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