Q1) What do you understand by Asymptotic notation, define different asymptotic notation with ~~exp~~ example

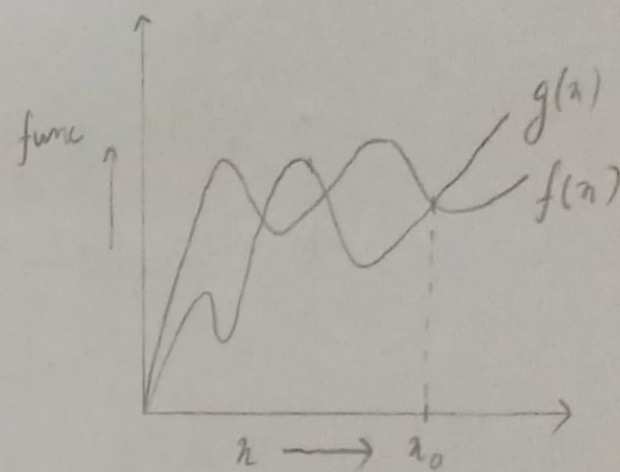
(i) Big O(n)

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

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

for some constant, $c > 0$

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



Eg. $f(n) = n^2 + n$

$g(n) = n^3$

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

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

(ii) Big Omega (Ω)

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

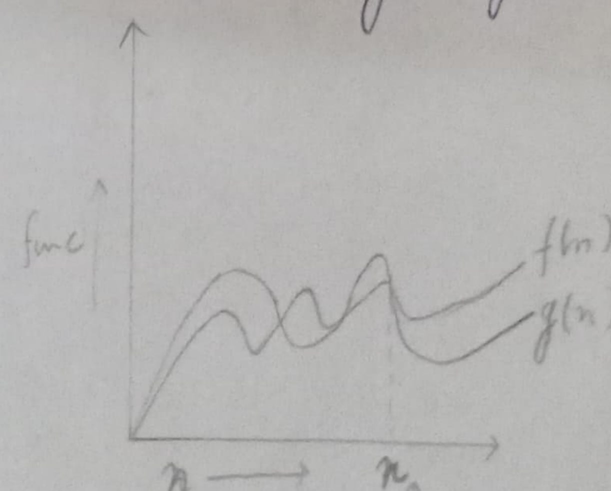
means $g(n)$ is 'tight' lowerbound of $f(n)$ ie $f(n)$ can go beyond $g(n)$

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

if & only if

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

$\forall \; n_2 > n_0$ & c: constant > 0



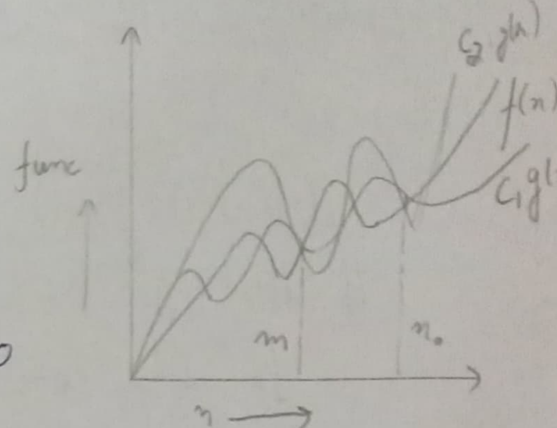
Eg. $f(n) = n^3 + 4n^2$
$g(n) = n^2$
ie $f(n) \geq c * g(n)$
$n^3 + 4n^2 = \Omega(n^2)$

iii) Big Theta (θ)

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

ie $f(n) = \theta(g(n))$
if & 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 $c_1 > 0$ & $c_2 > 0$

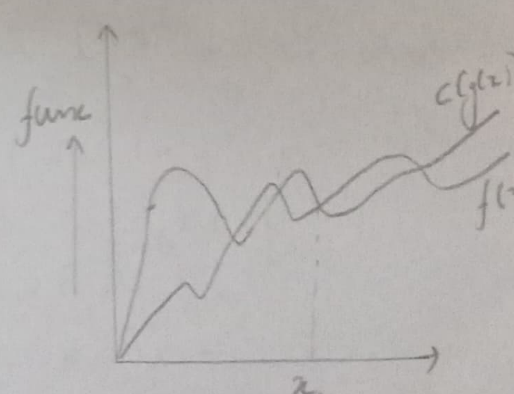


ie $f(n)$ can never go beyond $c_2(g(n))$ & will never come down of $c_1 g(n)$

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

(iv) Small O (o)

When $f(n) = o(g(n))$ gives the upper bound
ie $f(n) = o(g(n))$
if & ~~if~~ only if
$f(n) < c * g(n)$
$n^2 = o(n^3)$



(v) Small Omega (ω)

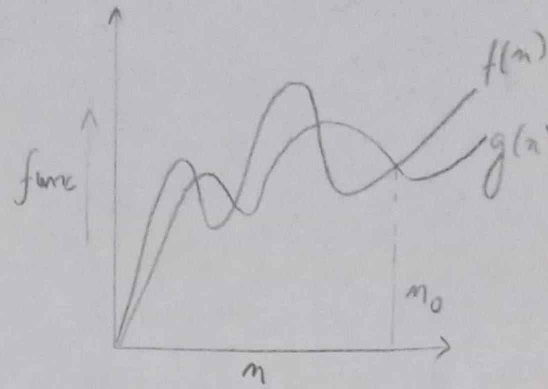
It gives the 'lower bound' ie

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

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

if & only if $f(n) > c * g(n)$

$\forall \; n > n_0$ & some constant, $c > 0$



Q2.) What should be time complexity of:

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

for $i = 1, 2, 4, 8, \ldots$ n times

ie series of GP.

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

K^{th} term of GP → $t_k = a r^{k-1}$

$t_k = 1(2)^{k-1}$

$2n = 2^{k-1}$

$\log_2(2n) = K \log 2$

$\log_2 2 + \log_2 n = k$

$\log_2 n + 1 = k$ (Neglecting '1')

So, time complexity $T(n) = O(\log n)$

4

Q3) $T(n) = \{ \ 3T(n-1)$ if $n>0$

otherwise 1

$\}$

i.e. $T(n) = 3T(n-1)$ — 1

$T(n) = 1$

put $n = n-1$ in 1

$T(n-1) = 3T(n-2)$ — 2

put 2 in 1

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

$T(n) = 9T(n-2)$ — 3

put $n = n-2$ in 1

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

put in 3

$T(n) = 27T(n-3)$ — 4

Generalizing,

$T(K) = 3^K T(n-K)$ — 5

for K^{th} terms, let $n-K=1$

$k = n-1$

Put in 5

$T(n) = 3^{n-1} T(1)$

$T(n) = 3^{n-1}$

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

Q4) $T(n) = \{ \quad 2T(n-1) - 1 \quad \text{if } n>0$

otherwise 0

}

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

put $n = n-1$

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

put in 1

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

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

put $n = n-2$ in 1

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

Put in 3

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

Generalizing,

$T(n) = 2^k T(n-K) - 2^{k-1} - 2^{k-2} \ldots 2^0$

K^{th} term →

let $n-K = 1$

$K = n-1$

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

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

in series in GP

$a = \frac{1}{2}, \quad r = \frac{1}{2}$

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

$= 2^{n-1} \left(\cancel{1} - \cancel{1} + \left(\frac{1}{2} \right)^{n-1} \right) = \frac{2^{n-1}}{2^{n-1}} = \quad T(n) = O(1)$

Q5.) What should be time complexity of

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

$i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad \ldots$

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

Sum $= 1+3+6+10 \ldots + n$

Also $s = 1 + 3 + 6 + 10 + \ldots + T_{n-1} + T_n$

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

$T_k = 1 + 2 + 3 + \ldots + k$

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

for k iterations

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

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

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

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

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

Q6) Time complexity of

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

As $i^2 = n$

$i = \sqrt{n}$

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

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

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

$$= \frac{n * \sqrt{n}}{2}$$

$T(n) = O(n)$

Q7) 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 (h=1 , h<=n ; k=k+2)
        count ++;
    }
```

Since, for $k = n^2$

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

$\because$ Series is in GP.

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

$$\frac{a(r^n - 1)}{r-1} = \frac{1(2^K - 1)}{2-1}$$

$n = 2^K - 1$

$n+1 = 2^k$

$\log_2 n = K$

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

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

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

Q8:) 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)
}
```

for (i=1 to n)

we get $j = n$ time every term

$\therefore i * j = n^2$

k^{th},

Now, $T(n) = n^2 + T(n-3)$

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

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

& $T(1) = 1$

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

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

Let

$$k^n - 3k = 1$$

$$k = (n-1)/3$$

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

$$T(n) \simeq k(n)^2$$

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

So,

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

Q9.) Time Complexity of

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

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

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

$$T(m) = 1 + d * m$$

$$(n-1)/d = m$$

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

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

$i=n-1$

we get,

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

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

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

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

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

Since $\int \frac{1}{n} = \log n$

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

Q10.) For the function n^k & C^n, what is asymptotic relationship b/w these function.
Assume that $k >= 1$ & $C > 1$ are constants. Find out the value of c & n_0 of which relationship holds.

As given n^k & c^n

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

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

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

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

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

$$\Rightarrow 1^k < a^2$$

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