Ques 1. T. C. $= O(n^2)$

↳ It takes 5 seconds for $n = 10$

Lets say $Kn^2 = 5$

$K(100) = 5$

$K = \frac{5}{100}$

for $n = 50$

↳ Time $= K(50)(50)$

$= \frac{5}{1\cancel{00}}(5\cancel{0})(5\cancel{0})$

$\Rightarrow$ 125 seconds

∴ approximately it will take 125 seconds

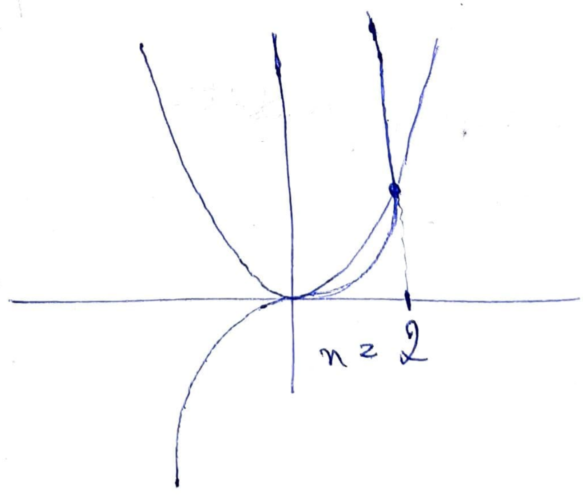
Ques 2. $T(A(n)) = n^3$

$T_{B(n)} = 2n^2$

So, $n^3 = 2n^2$

$n^2(n-2) = 0$

$\underline{(n = 2)}$



∴ after $n = 2$ they will start to deviate.

Ques 3. Use the limit rule to check that $n2^n$ is in $o(4^n)$

$$\lim_{n\to\infty} \frac{4^n}{n2^n}$$

$$\lim_{n\to\infty} \frac{2^{2n}}{n2^n}$$

$$\lim_{n\to\infty} \frac{2^n}{n}$$

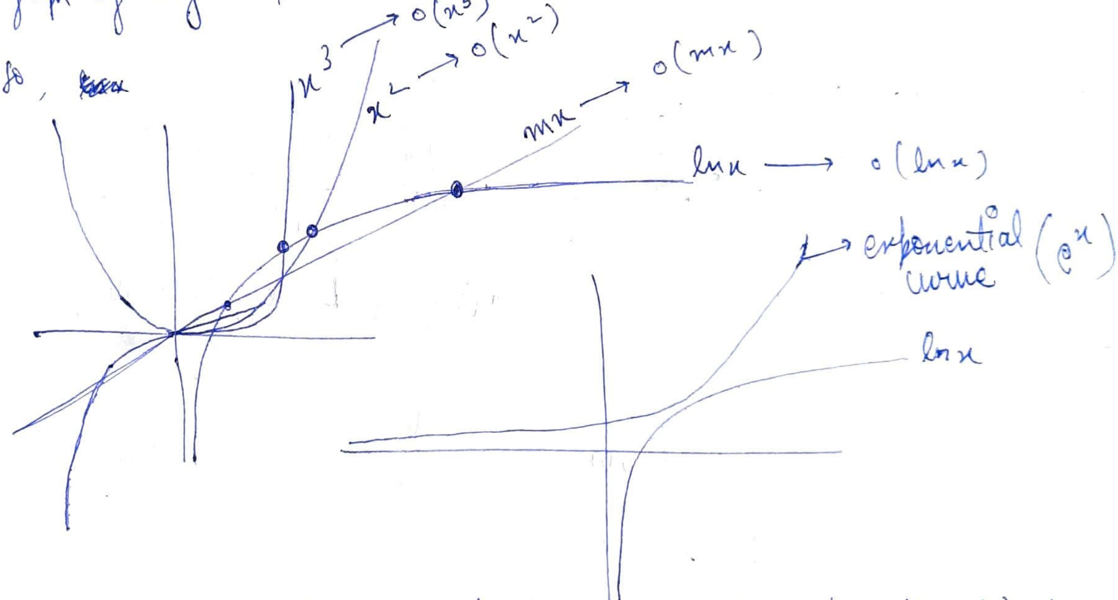
Applying L'hospital rule as $\frac{\to\infty}{\to\infty}$ form indeterminate

$$\lim_{n\to\infty} \frac{n2^{n-1}}{1} = \infty$$

$\therefore$ It means for large values of n 4^n is much bigger than $n2^n$ so $n2^n$ is in bounds of $o(4^n)$.

Ques 4. graph of $\log x$, Let assume base to e.

So,



$\therefore$ hence we can clearly see that for large value of x (input) log graph has slowest growth.

we can also see by differentiating them $\Rightarrow$

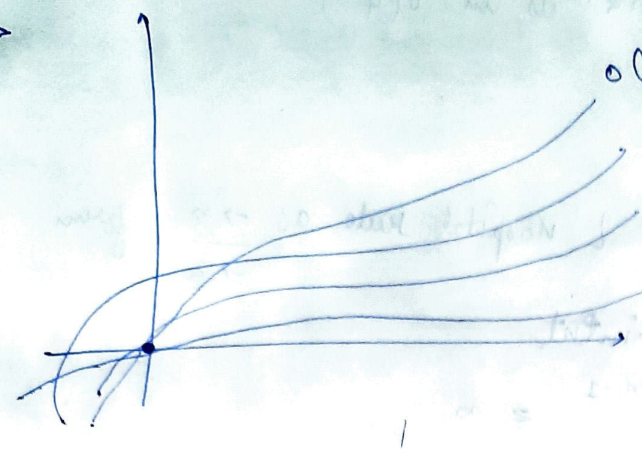
x^3	x^2	mx	e^x	$\log x$
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
$3x^2$	$2x$	m	e^x	$1/x$

Ques 5. a) θ and o

$\theta \to$ average case time complexity (theta)

↳ it is the avg of the time complexity that a f^n /algorithm performs, Eg $\to$ Quick sort has avg T. C $= \theta(n\log n)$ but has worst case Time complexity ie Big O $\approx o(n^2)$.

☆



$O(n) \longrightarrow$ worst time complexity.

$f(n)$

$\theta(n) \longrightarrow$ avg time complexity

$\Omega(n) \rightarrow$ omega of time complexity.

↓

It bounds the function from below.

☆. $\Omega(n)$ is shown in sorting algorithm like bubble sort when array already sorted fully.

<u>Ques 6.</u> (a) $n^4 + \log n + 17$

$$\lim_{n\to\infty} n^4 + \log n + 17$$

$$\lim_{n\to\infty} n^4\left(1 + \frac{\log n}{n^4}\!\!\xrightarrow{}0 + \frac{17}{n^4}\!\!\xrightarrow{}0\right)^{\infty}$$

$\because \lim_{n\to\infty} \frac{\log n \to \infty}{n^4 \to \infty}$

$\therefore$ applying L, Hospital rule,

$$\lim_{n\to\infty} \frac{\{n^4\}}{}$$

↓

$$\lim_{n\to\infty} \frac{\frac{1}{n}}{4n^3} = \frac{1}{4n^4} = 0$$

So for large n f^n behaves like $\underline{n^4}$ so, it has $O(n^4)$

Ques 7. a) K = 1

while K ≤ n

K = K+1

End while

T.C. = $O(n)$

(b) for i = 1 to n-1 do

for j = i+1 to n do

swap

End for

End for.

$$n + (n-1) + (n-2) + \cdots 2 + 1 - 1$$

$$\frac{n(n+1)}{2} \approx \boxed{O(n^2)}$$

Ques 8. algorithm T.C. $\Rightarrow O(n^2)$ it takes t_1 time for n input;

for

$2n$ inputs $(2n)^2 \Rightarrow 4n^2$

kn inputs $(kn)^2 = kn^2$

for $k = \sqrt{2}$

if input size increase by 1.42 times then that the time for running algorithm will become twice ie $2t_1$.

Ques 9. $T_A = ~~\text{xxxx}~~\ 100^n$

$T_B = n^4$

$$\lim_{n\to\infty} \frac{100^n \to \infty}{n^4 \to \infty} \Rightarrow \frac{n\,100^{n-1}}{4n^3} \Rightarrow \frac{(n-1)\,100^{n-2}}{8n} \Rightarrow \left[\frac{100^{n-2}}{8\left(\frac{1}{1-\frac{1}{n}\to 0}\right)}\right]_{\lim n\to\infty} \Rightarrow \infty$$

apply L'Hospital's rule.

So, T_A is much worse performing than T_B at larger values of n.

Ques 10. show that $n \log n \in \Theta(\log(n!))$

$$\lim_{n\to\infty} \frac{\log(n!)}{n\log n} \Rightarrow \frac{\log n}{n\log n} + \frac{\log n-1}{n\log n} + \frac{\log n-2}{n\log n} + \dots \ 0 = 0$$

$$\lim_{n\to\infty} \frac{n\log n}{\log n!} \Rightarrow \frac{\log n^n}{\log n!} \Rightarrow \lim_{n\to\infty} \frac{n^n}{n!} \Rightarrow \frac{n\times n\times n\times\dots}{n(n-1)(n-2)\dots 1}$$

we can do this because log is continuous always increasing f^n.

$$\Rightarrow \frac{1}{\left(1-\frac{1}{n}\right)^0\left(1-\frac{2}{n}\right)^0\left(1-\frac{3}{n}\right)^0\dots}$$

$\Rightarrow 1$

hence proved that $n\log n \simeq \Theta(\log(n!))$

Ques 11. (a) $2^{n-1} + 4^{n+1}$

$$\frac{2^n}{2} + 2^{2n+2}$$

$$\frac{2^n}{2} + 4\,2^{2n}$$

$$2^n \underbrace{\left(\frac{1}{2} + 4\,2^n\right)}$$

$\simeq 2^n$] for $n \to \infty$

$$2^{2n} = 4^n$$

$\Rightarrow O(4^n)$

(b) $(n^2+6)^8$

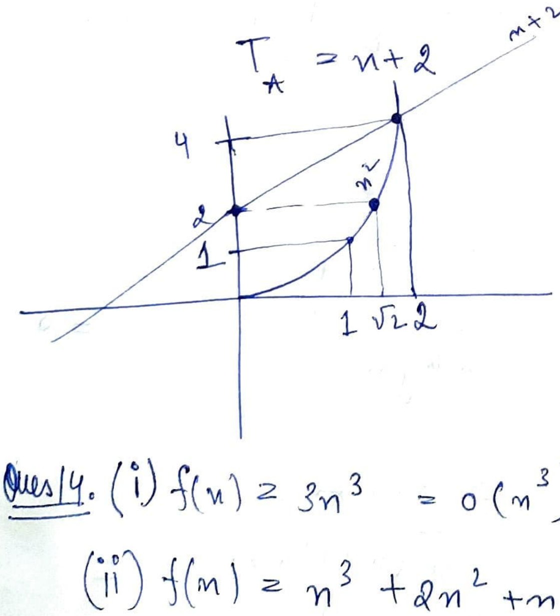
$$\lim_{n \to \infty} \to (n^2)^8 \left(1 + \frac{6}{n^2}\right)^8$$

$= n^{16}$

$\Rightarrow \simeq O(n^{16})$

Ques 12. $T_A = n^2$

$T_A = n+2$



breaking point is $n=2$ after $n=2$ n^2 grows much faster than $n+2$.

Ques 14. (i) $f(n) = 3n^3 = O(n^3)$

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

(iii) $f(n) = 2n^3 + 13\log_2 n \,/\, 7n^2 \Rightarrow \frac{2n}{7} + \frac{3\log n}{n^2} = O(n)$

* for (i=1 ; i ≤ n ; i++)

{

$= O(n)$

}

* for (i=n-1 , i ≥ 1 ; i /= 2)

$\Rightarrow O(\log_2 n)$

* for (i=0 ; i < n , i++) $\rightarrow O(n)$

for (j=0 , j < n, j++) $\rightarrow O(n)$

$\Rightarrow O(n^2)$

* for (i=0; i < n ; i++) $\rightarrow O(n)$

for (i=n-1 ; i ≥ 1 ; i = i/2) $\rightarrow O(\log_2 n)$

$\Rightarrow O(n \log n)$

* for (i=0 ; i < n ; i++) $\rightarrow O(n)$

for (j=0 ; j < n; j++) $\rightarrow O(n)$

for (k=0 ; k < n ; k++) $\rightarrow O(n)$

$O(n^3)$

* for (i=0 ; i < n ; i++)

for (j=0 ; j ≤ i ; j++)

$\Rightarrow O(n^2)$

* for (i=0; i<n-1; i++) → O(n)

 for (j=i+1; j<n; j++) →

$\Rightarrow O(n^2)$

* for (x=n-2; x≥0; x--)

 for (j=0; j≤x; j++)

$\approx O(n^2)$

* i=1; s=1

 while (s<=n)

 i++;

 s=s+i;

* for (i=n/2; i<=n; i++)

 for (j=1; j<n; j=2*j)

 for (k=1; k≤n; k=2*k)

$O(n(\log_2 n)^2)$

growth of s is in order of $\underbrace{n^2}$ so time complexity would be $\sqrt{n}$.

$\Downarrow$

as an^2+bn

* for(i=1; i≤n; i++) →

 for (j=1; j≤i; j++)

 { for(k=1; k≤100; k++)

 {

 } }

$\approx O(n^2)$

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

 for (j=1; j<=i^2; j++)

 for (k=1, k≤n/2; k++)

 → $O(n)$

$O(n) \times O(n^3) = O(n^4)$

* for (i=n/2; i<=n; i++)

 for (j=1; j≤n/2; j++)

 for (k=1; k<=n; k=k*2)

$O(n^2 \log_2 n)$