Design Algorithm Of Analysis

Theory Assignment :-

① What is an algorithm? explain various characteristics of an algorithm.

Ⓐ Algorithm is Nothing but step by step process for solving a problem.

* Abstract computation procedure which takes some values as input and produce a value(s) as output.
* We will write an algorithm at design time.
* It is independent of platform.
* It depends on domain knowledge.
* We analyse the algorithm function of the time
* It is an priori testing

Characteristics of an algorithm:

1) Input : The algorithm must have 0 (or) More inputs.
2) Output : It must have atleast one output
3) Finiteness: It must terminate after a finite no. of steps.
4) Definiteness: Each step of algorithm must be clear
5) Effectiveness : Each step of algorithm must be correct and it should happen in finite count of time.

Scanned with CamScanner

② List and explain various ways of expressing language. an algorithm.

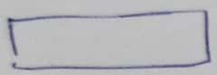
Ⓐ There are three ways of expressing an Algorithm.

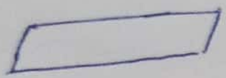
1) Natural language
2) pseudo code
3) Flow chart

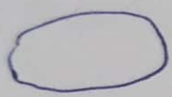
Natural language: It is low level language which is platform independent and can be written in Simple english but it has many drawbacks associated with it like unambiguous, do not have proper structure and dificult to find its location changes

Pseudo Code: It uses programing constructs to write an algorithm and it is an platform independent The pseudo code has an advantage of being easilly converted into any programming Language This pseudo code is language independent and it has no standard of writing it.

Flow chart: Flow chart is an pictorial representation of an algorithm. It is easy to understand a flow chart. Flow chart has a simple geometric shapes to depict processes and arrows to relationships and dataflow. some Geometric shapes include

 → Rectangle → steps of algorithm

 → parallelogram → Input and output

 → oval → start/end.

 → Rhombus → condition/selection.

(Q) What is a recurrance relation? How to solve recurance relation?

(A) <u>Recurance Relation</u>: When an algorithm contains a recursive call to itself then its time complexity can be described by recurence relation.

There are three methods to solve recurance relations:

1) Substitution Method (Forward and backward sub^n)
2) Recurrance tree Method.
3) Master method.

<u>Substitution Method</u>:

This method has two types forward substitution and backward substitution.

<u>Example</u>:

(1) Find the boundry in the first step.

$$T(n) = \begin{cases} 1 & , n = 0 \\ T(n-1)+1 & , \text{otherwise } n > 0 \end{cases}$$

(2) Now start with T(n) and expand it recursively using recurrance.

Scanned with CamScanner

$T(n) = T(n-1) + 1 \longrightarrow ①$

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

Substitute ② in ①

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

$T(n) = T(n-2) + 2 \longrightarrow ③$

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

Substitute ④ in ③.

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

$T(n) = T(n-3) + 3 \longrightarrow ⑤$

$\vdots$

$T(n) = T(n-k) + k \longrightarrow ⑤$

if $n-k=0$.

$\Rightarrow n = k$.

Sub $k=n$ in ⑤

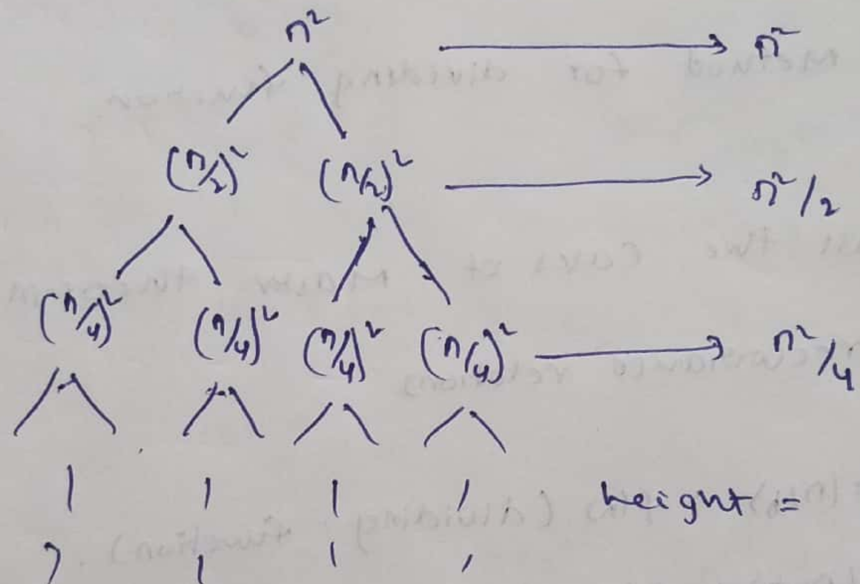
$T(n) = T(0) + n$.

$T(n) = O(n)$.

$\therefore$ The time complexity of $T(n)$ is $O(n)$.

Recurrance tree Method:

It is also known as graph Method. In recursion tree method we should find the cost of each level, we sum the cost with each of the level of the tree to obtain a set of pre-level costs and then sum all pre-level costs to determine the total cost of all levels of the recurssion. and then find height of the tree.



height $= \left(\frac{n}{2^k}\right)^2$

$\boxed{k = \log n.}$

$$T(n) = n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \cdots + \frac{n^2}{2^k}$$

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

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

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

Since we consider in growth.

$\boxed{T(n) = O(n^2).}$

Master Method:

There are two ways to solve Masters method.

1) Master Method for reducing. (Substraction function).

2) Master Method for dividing function.

⑥ Explain all the cases of Master theorem for the following recurrance relations

$T(n) = aT(n/b) + f(n)$ (dividing function).

$T(n) = T(n-b) + f(n)$ (decreasing function).

Decreasing function:

$$T(n) = aT(n-b) + f(n).$$

where, $a>0$, $b>0$, $f(n) = O(n^k)$.

if $a=1$, $T(n) = O(n * f(n))$

$a>1$, $T(n) = O(a^n * f(n))$

$a<1$, $T(n) = O(f(n))$.

Dividing function:

$T(n) = a\,T(n/b) + f(n).$

Where, $a >= 1$, $b > 1$, $f(n) = O(n^k \log^p n)$

Case 1: if $\log_b^a > k$, then $O(n^{\log_b^a})$

Case 2: if $\log_b^a = k$.

(i) if $p > -1$ them $O(n^k \log^{p+1} n)$

(ii) if $p = -1$ then $O(n^k \log\log n)$

(iii) if $p < -1$ then $O(n^k)$.

Case 3: if $\log_b^a < k$

(i) if $p >= 0$ $O(n^k \log^p n)$

(ii) if $p < 0$ $O(n^k)$

(7) Solve the following Recurrence relations using Substitution, recurrence tree and Master method.

(1) $T(n) = 7T(n/2) + 3n^2 + 2$

$$T(n) = \begin{cases} 1 & , \text{if } n=1 \\ 7T(n/2) + 3n^2 + 2 & , \text{otherwise} \end{cases}$$

$T(n/2) = 7T(n/4) + 3(n/2)^2 + 2$

$T(n) = 7(7T(n/4) + 3(n/2)^2 + 2) + 3n^2 + 2$

$= 7^2 T(n/4) + 7 \cdot 3 \cdot (n/2)^2 + 7 \cdot 2 + 3n^2 + 2$

$T(n) = 7^k T(n/2^k) + 3n^2\left[1 + \frac{7}{4} + \frac{7^2}{4^2} + \cdots\right] + 2[1+7+7^2+\cdots]$

$T(n) = 7^k T(n/2^k) + 3n^2\left(\frac{(7/4)^k - 1}{(7/4) - 1}\right) + 2\left(\frac{7^k - 1}{7 - 1}\right)$

if $n/2^k = 1 \Rightarrow \boxed{k = \log n}$

$T(n) = 7^{\log n} T(n/2^k) + 3n^2\left(\frac{(7/4)^{\log n} - 1}{7/4 - 1}\right) + 2\left(\frac{7^{\log n} - 1}{6}\right)$

$T(n) = n^{\log 7}(1) + 4n^2\left((7/4)^{\log n} - 1\right) + \left(\frac{n^{\log 7} - 1}{3}\right)$

$T(n) = n^2 \cdot n^{\log 7/4}$

$T(n) = n^{2 + \log 7 - \log 4}$

$\boxed{T(n) = O(n^{\log 7})}$

② $T(n) = 2T(n/2) + n^2$

$a = 2,\ b = 2,\ f(n) = n^2,\ k = 2,\ p = 0$

$\log_b a = \log_2 2 = 1 \Rightarrow 1 < k$

$\because \log_b a < k$ and $p = 0$

$T(n) = O(n^k \log^p n)$

$\boxed{T(n) = O(n^2)}$

④ $T(n) = 4T(n/2) + n^3$

$a = 4,\ b = 2,\ f(n) = n^3,\ p = 0,\ k = 3$

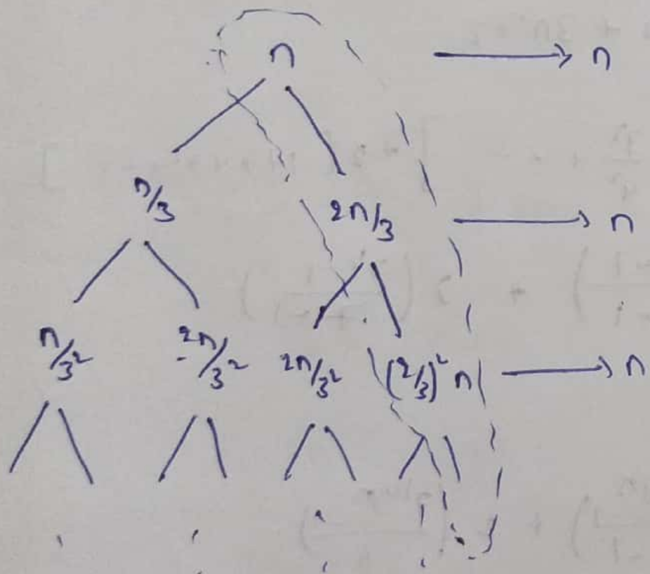
$\log_b a = \log_2 4 = 2 < k$

$\because \log_b a < k$ & $p = 0$

$T(n) = O(n^k \log^p n)$

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

③ $T(n) = T(n/3) + T(2n/3) + n$



$n,\ 2n/3,\ \left(\frac{2}{3}\right)^2 n,\ \ldots,\ \left(\frac{2}{3}\right)^k n.$

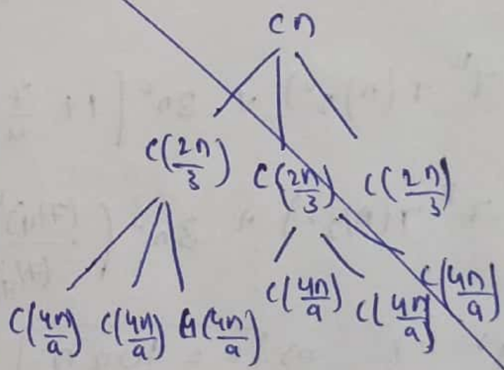
if, $\left(\frac{2}{3}\right)^k n = 1 \Rightarrow n = (2/3)^{-k}$

$\Rightarrow \boxed{k = \log_{3/2} n}$

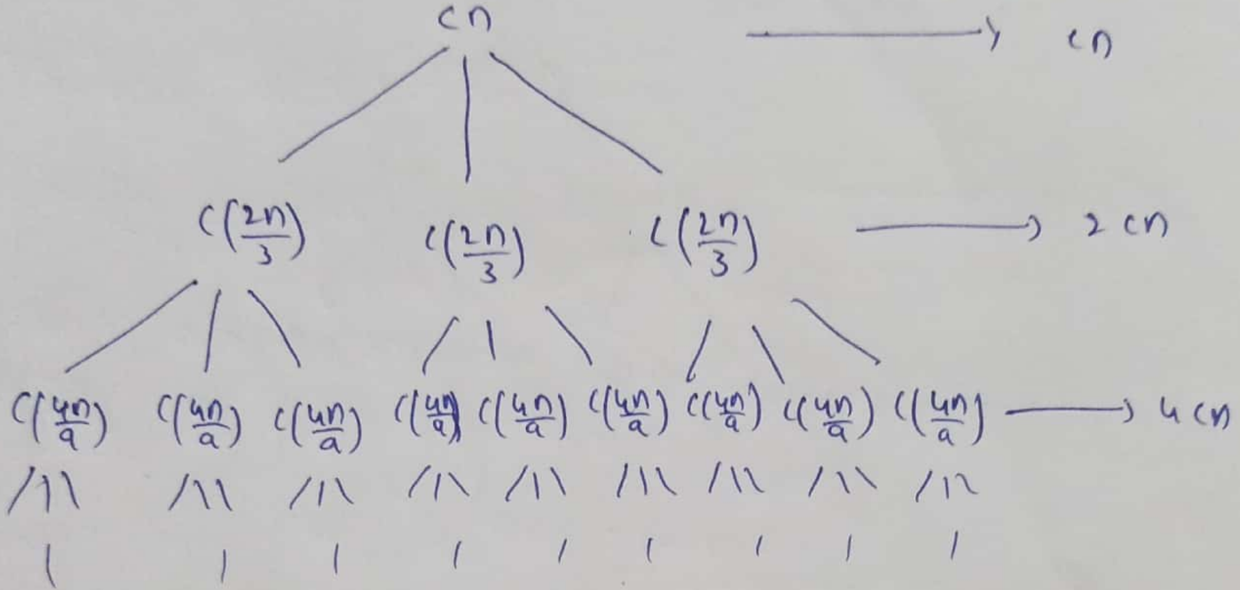
$T(n) = kn$

$\boxed{T(n) = O(n \log_{3/2} n)}$

⑤ $T(n) = 3T(2n/3) + cn$



⑤ $T(n) = 3T(2n/3) + cn$



$cn, \frac{2}{3} cn, \frac{4}{9} cn, \ldots \left(\frac{2}{3}\right)^k cn$

if $\left(\frac{2}{3}\right)^k cn = 1$

$cn = \left(\frac{3}{2}\right)^k$

$\log cn = k \log 3/2$

$\boxed{k = \log_{3/2} cn}$

$T(n) = cn + 2cn + 4cn + \cdots + 2^k cn$

$T(n) = cn\,(1 + 2 + + \cdots + 2^k)$

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

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

$T(n) = cn\, 2^{\log_{3/2} n} - cn$

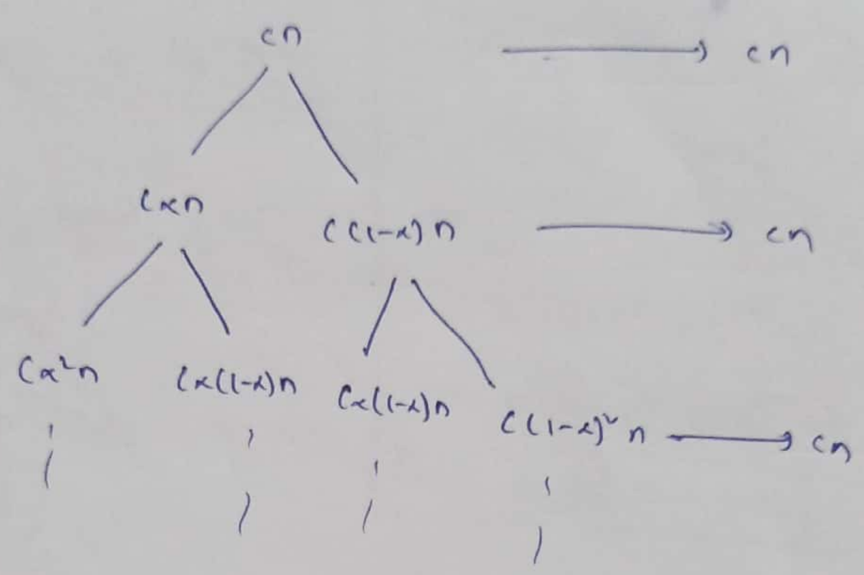
$T(n) = cn\, n^{\log_{3/2} 2}$

$T(n) = (cn)(n^{\log_{3/2} 2}) - cn$

$\boxed{T(n) = O(n^{1 + \log_{3/2} 2})}$

6. $T(n) = T(n-1) + T(n/2) + n$

(7) $T(n) = T(\alpha n) + T((1-\alpha)n) + cn.$



$cn,\ c(1-\alpha)n,\ c(1-\alpha)^2 n, \ldots, c(1-\alpha)^k n.$

$c(1-\alpha)^k n = 1$

$\log c(1-\alpha)^k + \log n = 0$

$\log cn + k \log(1-\alpha) = 0.$

$\log cn = -k \log(1-\alpha)$

$$\boxed{k = \log_{cn}(1-\alpha)}$$

$T(n) = k(cn)$

$T(n) = (c)\, n \log_{cn} 1-\alpha$

$$\boxed{T(n) = O(n \log_{cn}(1-\alpha))}$$

(8) $T(n) = T(n-a) + T(a) + cn.$

(9) $T(n) = 2T(n/4) + 1$

$a=2, b=4, f(n)=1, k=0, p=0$

$\log_b a = \log_4 2 = 1/2 > k$

$\because \log_b a > k$ and $p=0$

$T(n) = O(n^{\log_b a})$

$T(n) = O(n^{\log_4 2})$

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

(10) $T(n) = 2T(n/4) + \sqrt{n}$

$a=2, b=4, f(n)=\sqrt{n}, k=1/2, p=0$

$\log_b a = \log_4 2 = 1/2 = k$

$\because \log_b a = \log_4 2 = 1/2 = k$ and $p=0$

So

$T(n) = O(n^k \log^{p+1} n)$

$T(n) = O(n^{1/2} \log^{0+1} n)$

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

(11) $T(n) = 2T(n/4) + n$

$a=2, b=4, f(n)=n, k=1, p=0$

$\log_b a = \log_4 2 = 1/2 < k$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^{1/2} \log^0 n)$

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

(12) $T(n) = 2T(n/4) + n^2$

$a=2, b=4, f(n)=n^2, k=2, p=0$

$\log_b a = \log_4 2 = 1/2 < k$

$\because \log_b a < k$ and $p=0$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^2 \log^0 n)$

$\boxed{T(n) = O(n^2)}$

(13) $T(n) = 4T(n/2) + n^2 \log n$

$a=4, b=2, f(n)=n^2 \log n, p=1, k=2$

$\log_b a = \log_2 4 = 2 = k$

$\because \log_b a = k$ and $p=1$

$T(n) = O(n^k \log^{p+1} n)$

$\boxed{T(n) = O(n^2 \log^2 n)}$

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

$a=1, a>0, b>0, f(n)=n$

$\because a=1,$

$T(n) = O(n * f(n))$

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

$\boxed{T(n) = O(n^2)}$

15

16

17 $T(n) = 4T(n/2) + n^2;$

$a=4,\ b=2,\ f(n)=n^2,\ k=2,\ p=0$

$\log_b a = \log_2 4 = 2 = k$

$\because \log_b a = k$ and $p > -1$

$T(n) = O(n^k \log^{p+1} n)$

$T(n) = O(n^2 \log^{0+1} n)$

$\boxed{T(n) = O(n^2 \log n)}$

18

19 $T(n) = 2T(n/2) + n^4$

$a=2,\ b=2,\ k=4,\ p=0,\ f(n)=n^4$

$\log_b a = \log_2 2 = 1 < k.$

$\because \log_b a < k$ and $p=0$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^4 \log^0 n)$

$\boxed{T(n) = O(n^4)}$

20 $T(n) = T(7n/10) + n$

$a=1,\ b=10/7,\ f(n)=n,\ k=1,\ p=0$

$\log_b a = \log_{10/7} 1 = 0 < k$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^1 \log^0 n)$

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

21 $T(n) = 16T(n/4) + n^2$

$a=16,\ b=4,\ f(n)=n^2,\ k=2,\ p=0$

$\log_b a = \log_4 16 = 2 = k$

$\because \log_b a = k$ and $p > -1$

$T(n) = O(n^k \log^{p+1} n)$

$T(n) = O(n^2 \log^{0+1} n)$

$\boxed{T(n) = O(n^2 \log n)}$

(18) $T(n) = 3T(\sqrt{n}) + \log n$

$$T(n) = \begin{cases} 1, & \text{if } n=2 \\ 3T(\sqrt{n}) + \log n, & \text{otherwise} \end{cases}$$

$T(\sqrt{n}) = 3T(n^{1/4}) + \log(\sqrt{n})$

$T(n) = 3(3T(n^{1/4}) + \log\sqrt{n}) + \log n.$

$T(n) = 3^2 T(n^{1/4}) + 3\log\sqrt{n} + \log n$

$T(n) = 3^k T(n^{1/2^k}) + \log n . n^{3/2} . n^{3^2/4} \ldots n^{3^k/2^k}$

$T(n) = 3^k T(n^{1/2^k}) + (1 + 3/2 + 3^2/4 + \ldots + 3^k/2^k)\log n.$

$$T(n) = 3^k T(n^{1/2^k}) + \left(\frac{(3/2)^k - 1}{3/2 - 1}\right)\log n$$

$T(n) = 3^k . T(n^{1/2^k}) + 2(3/2)^k \log n - 2\log n.$

if $n^{1/2^k} = 2$

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

$\Rightarrow \boxed{\log(\log n) = k}$

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

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

$\boxed{T(n) = O((\log n)^{1+\log 3/2})}$. (∵ We consider growth).

(22) $T(n) = 7T(n/3) + n^2$

$a=7,\ b=3,\ k=2,\ f(n)=n^2,\ p=0$

$\log_3 7 = 1.771 < k$

$\because \log_b a < k$ and $p=0$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^2 \log^0 n)$

$\boxed{T(n) = O(n^2)}$

(23) $T(n) = 7T(n/2) + n^2$

$a=7,\ b=2,\ f(n)=n^2,\ k=2,\ p=0$

$\log_b a = \log_2 7 = 0.8 < k$

$\because \log_b a < k$, and $p=0$

$T(n) = O(n^k \log^p n)$

$T(n) = O(n^2 \log^0 n)$

$\boxed{T(n) = O(n^2)}$

(24) $T(n) = 2T(n/4) + \sqrt{n}$

$a=2,\ b=4,\ f(n)=\sqrt{n},\ k=1/2,\ p=0$

$\log_b a = \log_4 2 = 1/2 = k$

$\because \log_b a = k$ and $p > -1$

$T(n) = O(n^k \log^{p+1} n)$

$T(n) = O(n^{1/2} \log^1 n)$

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

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

$a=1,\ f(n)=n^2$

$T(n) = O(n \times f(n))$

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

Scanned with CamScanner

⑧ Group the functions so, that $f(n)$ and $g(n)$ are in same group if $f(n) = O(g(n))$ and $g(n) = O(f(n))$. List the group in increasing order.

$\sqrt{n}$	n	2^n
$n\log n$	$n - n^3 + 7n^5$	$n^2 + \log n$
n	n^3	$\log n$
$n^{1/3} + \log n$	$\log^2 n$	$n!$
$\ln n$	$n/\log n$	$\log\log n$
$(1/3)^n$	$(2/3)^n$	6

Ⓐ Time complexities:

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

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

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

$n^{1/3} + \log n = O(n^{1/3})$

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

$(1/3)^n = O((1/3)^n)$

$n = O(n)$

$n - n^3 + 7n^5 = O(n^5)$

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

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

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

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

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

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

$n! = O(n!)$

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

$6 = O(1)$

$n/\log n = O(n/\log n)$.

Scanned with CamScanner

$O(1) \longrightarrow 6$

$O(\log\log n) \longrightarrow \log(\log n)$

$O(\log n) \longrightarrow \log n$

$\ln n \quad (\because \ln n \sim \log n)$

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

$O((1/3)^n) \longrightarrow (1/3)^n$

$O(n^{1/3}) \longrightarrow n^{1/3} + \log n$

$O(n^{1/2}) \longrightarrow \sqrt{n}$

Increasing order.

$O(n/\log n) \longrightarrow n/\log n$

$O(n) \longrightarrow n$

$O(n\log n) \longrightarrow n\log n$

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

$n^2 + \log n, \ldots$

$O(n^3) \longrightarrow n^3$

$O(n^5) \longrightarrow n^5$

$\cancel{O((1/3)^n) \longrightarrow (1/3)^n}$

$O((3/2)^n) \longrightarrow 2^n$

$O(n!) \longrightarrow n!$

Order of time complexity:

$O(1) < O(\log\log n) < O(\log n) < O(\log^2 n) \overset{O((1/3)^n)}{<} O(n^{1/3}) < O(\sqrt{n})$

$< O(n/\log n) < O(n) < O(n\log n) < O(n^2) < O(n^3) < O(n^5) \cancel{< O((1/3)^n)}$

$< O((3/2)^n) < O(2^n) < O(n!)$

(10) Consider the following fourteen functions for the equations that follows

(a) $\log_3(2n)$

(b) $\sqrt{n}$

(c) $n\log_3(n/2)$

(d) $\log_2 3n^2$

(E) 2^n

(F) 2^{n+2}

(G) 2^{2n}

(H) $3n+5\log_2(n)$.

(I) $5n+\sqrt{n}$.

(J) $\sum_{k=1}^{n} k = 1+2+3+4+ \cdots + n.$

(k) $\sum_{k=1}^{n^2} k = 1+2+3+4+ \cdots + 2n$

(l) $\sum_{k=1}^{n^2} k = 1+2+3+4+ \cdots + n^2$

(m) $\sum_{k=1}^{n} k^3 = 1^3+2^3+3^3+4^3+ \cdots + n^3$

(n) $\sum_{k=1}^{n} 2^k = 1+2+4+8+ \cdots + 2^n.$

Make a table in which each function is in a column dictated by its Θ growth rate. Functions which with the Same asymptotic growth rate should be ordered left to right by the rate of growth of their functions. columns with slower growing functions should be to the left of colums with faster growing functions.

(18) (A)

(a) $\log_3 2n = \log_3 2 + \log_3 n = O(\log n)$

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

(c) $n \log_3 n/2 = O(n \log n)$.

(d) $\log_2 3n^2 = 2 \log_2 3n = 2\log_2 3 + 2\log_2 n = O(n \log n)$

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

(f) $2^{n+2} = O(2^n)$

(g) $2^{2n} = 4^n$

(h) $3n + 5\log_2 n = O(n)$

(i) $5n + \sqrt{n} = O(n)$

(j) $\sum_{k=1}^{n} k = 1+2+3+ \cdots + n = n(n+1)/2 = O(n^2)$

(k) $\sum_{k=1}^{2n} k = 1+2+3+ \cdots + 2n = n(n+1)/2 = O(n^2)$

(l) $\sum_{k=1}^{n^2} k = 1+2+3+ \cdots + n^2 = n(n+1)/2 = O(n^2)$.

(m) $\sum_{k=1}^{n} k^3 = 1^3+2^3+3^3 + \cdots + n^3 = n^2(n^2+1)/2 = O(n^4)$.

(n) $\sum_{k=1}^{n} 2^k = 2^0 + 2^1 + \cdots + 2^k = 1(2^n-1)/2-1 = O(2^n)$.

$O(\log n) \longrightarrow \log_3 2n$

$\log_{\frac{3}{2}} n^2$

$O(\sqrt{n}) \longrightarrow \sqrt{n}$

$O(n) \longrightarrow 3n + 5\log_2(n)$

$5n + \sqrt{n}$

$O(n \log n) \longrightarrow n \log_3 n/2$

$O(n^2) \longrightarrow \sum_{k=1}^{n} k, \sum_{k=1}^{2n} k, \sum_{k=1}^{n^2} k$

Increasing order.

$O(n^4) \longrightarrow \sum_{k=1}^{n} k^3$

$O(2^n) \longrightarrow \sum_{k=1}^{n} 2^k,$

$2^n, 2^{n+2}.$

$O(4^n) \longrightarrow 2^{2n}.$

$\therefore$ order of Time complexity :

$O(\log n) < O(\sqrt{n}) < O(n) < O(n\log n) < O(n^2) < O(n^4) < O(2^n) < O(4^n)$