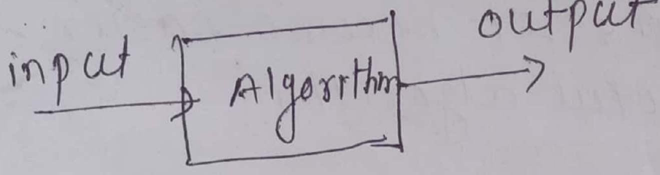
Unit-1

Q:- Define algorithm in detail. Explain their four distinct area of study. (W-18/6M)

Ans:-

Informal defination of algorithm:-

An algorithm is a well defined computational procedure that takes some values or set of values as Input and produces a set of values or some value as output.



Formal defination

An algorithm is a finite set of Instructions that if followed, accomplishes a particular task.

All algorithm must have following characteristic :-

1) Finiteness :- Algorithm must complete after a finite number of instructions have been executed.

2) Input :- zero or more external quantities must be supplied

3) Output :- Atleast one quantity is produced

4) Definiteness :- Each enstruction is clear and unambiguous

5) Effectivness :- Every instruction must very basic.

There are four distinct areas of study algorithm

1) How to devise algorithms
2) How to validate algorithms
3) How to analyze algorithms
4) How to test a program

How to devise algorithms :-

By studying various techniques i.e design strategies it become easier to devise and useful algorithm.

How to validate algorithms :-

Once an algorithm is devised, it is necessary to show that it computes the correct answer for all possible legal inputs.
Methods used for validation include
- contradiction method
- Mathmatical Induction

How to analyze algorithms :-

Analysis of algorithms or performance analysis refer to the task of determining how much computing time and storage an algorithm requires. We analyze the algorithm based on time & space complex

The amount of time needed to run the algorithm is called time complexity.

The amount of of memory needed to run the algorithm is called space complexity.

How to test a program :-

Testing a program ~~contains~~ consists of two phases.

1) Debugging 2) Profiling

Debugging :- It is the process of executing programs on sample data sets to determine wheather faulty results occur and if so to correct them.

Profiling :- Profiling or performance measurement is the process of executing a correct program on data sets and measuring the time and space it takes to compute the results.

Que:- Explain algorithm design strategy in de
(CS-17)
Ans:- for a given problem, there are many wa to design alogorithms for it.
Following is are the algorithm design strategies
1) Divide and conquer (D and C)
2) Greedy approach
3) Dynamic programming
4) Branch and Bound
5) Randomized algorithms
6) Backtracking algorithms

Divide & conquer:-
- Divide the original problem into a se of subproblems.
- Solve every subproblem individuall recursively.
- Combine the solutions of the sub problems (top level) into a solutio of the whole original problem.
eg. Binary search, Merge sort

Greedy Approach:-
Greedy algorithms seek to optimize a function by making choices (greed criterion) which are the best locally but do not look at the global probl

The result is a good solution but not necessarily the best one.

- The greedy algorithm does not always guarantee the ~~best one~~ optimal solution however it generally produces solutions that are very close in value to the optimal. eg. Minimum spanning tree algo.

Dynamic programming :-

- It is a technique for efficiently computing recurrences by storing partial results.
- It is a method of solving problems exhibiting the properties of overlapping subproblems and optimal substructure ~~that~~ takes much less time.

eg. Longest Common Subsequence, matrix-chain multiplication

Branch and Bound :-

- a given subproblem, which cannot be bounded, has to be divided into at least two new restricted subproblems.
- Branch and bound algorithms are methods for global optimization in nonconvex problems.
- It can be slow algorithm technique.

eg. 0/1 knapsack problem.

Randomized Algorithms :-

- A randomized algorithm is defined an algorithm that is allowed to ac a source of endependent, unbiased random bits, and it is then allow to use these random bits to influen its computation. (Ex Randomized Quick Sort we use random number to pick the next pivot

Backtracking Algorithms :-

- Backtracking algorithms try each possibility until they find the right
- It is a depth-first search of the set of possible solutions.
- During the search, if an alternative doesn't work, the search backtracks to the choice point, the place which presented different alternatives and tries the next alternatives.
- When the alternatives are exhaused, the search returns to the previous choice point and try the next alternative there
- If there are no more choices points, the search fails.

eg. 4 Queen problem, 8 Queen problem, Graph Coloring

Que:- Differentiate between recursive and iterative algorithm design. (3-18/3M)

ans:-

Recursive Algorithm	Iterative algorithm
1) In recursive approach, function calls itself until the condition is met	1) In iterative approach, function repeats until the condition fails
2) It uses a branching structure	2) It uses a looping construct.
3) Recursive solutions are often less efficient in terms of time and space	3) Iterative algorithm are more efficient
4) Recursion terminates when a base case is recognized	4) Iteration terminates when the loop continuation condition fails.
5) Infinite recursion occurs if the recursion step does not reduce the problem in a manner that converges on the base case	5) An infinite loop occurs with iteration if the loop continuation test never becomes false.
6) size of code is comparitively smaller in recursion.	6) Iteration makes the code size bigger

Recursion Example

C program to find factorial of a number using recursion

```
# include <stdio.h>
int factorial(int n)
{ if (n==0)
        return 1;
  else
     return n * factorial(n-1);
}
int main()
{
 printf("factorial for 5 is %d", factorial);
 return 0;
}
```

Iteration Example

```
# include <stdio.h>
int main()
{
  int i, n=5, fact=1;
  for(i=1; i<=n; i+1)
   fact = fact * i;
  printf("factorial for 5 is %d", fact);
  return 0;
}
```

Q:- Solve the following recurrance (W-17/7M)

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

$= 2t_{n-1} + 2^n + 5$ otherwise

A:- Rewrite the recurrance as

$t(n) - 2t_{n-1} = 2^n + 5$ —— (1)

Compare recurrance with

$a_0 t_n + a_1 t_{n-1} + a_2 t_{n-2} \cdots = b_1^n P_1(n) + b_2^n P_2(n)$

$\therefore k = 1,\ a_0 = 1,\ a_1 = -2,$

$b_1 = 2,\ P_1(n) = \cancel{4}\ 1$ which is a polynomial of degree 0.

$d_1 = 0$

$b_2 = 1,\ P_2(n) = 5$ which is a polynomial of degree 1

$d_2 = 1$

$\therefore$ Characteristic eq^n is

$(a_0 x^k + a_1 x^{k-1} + \cdots a_k)(x-b)^{d+1}$

$(a_0 x^k + a_1 x^{k-1} + \cdots a_k)(x-b_1)^{d_1+1}(x-b_2)^{d_2+1}$

$\therefore$ Characteristic eq^n is

$(x-2)(x-2)(x-1)^2$

$r_1 = 2,\ r_2 = 2,\ r_3 = 1,\ r_4 = 1$

$\therefore$ General solution is of the form

$t_n = c_1 r_1^n + c_2 n r_2^n + c_3 r_3^n + c_4 n r_4^n$

Scanned by CamScanner

$\therefore t_n = c_1 2^n + c_2 n 2^n + c_3 1^n + c_4 n 1^n$

put $n = 0$ in eq^n (A) we get

$\therefore \quad c_1 + c_3 = 3 \quad —— (i)$

put $n = 1$ in eq^n (1) we get

$t_1 - 2t_0 = 2 + 5$

$t_1 - 2 \times 3 = 7$

$t_1 - 6 = 7$

$\boxed{t_1 = 1}$

put $n = 1$ in eq^n (A) we get

$2c_1 + 2c_2 + c_3 + c_4 = 1 \quad —— (ii)$

put $n = 2$ in eq^n (1) we get

$t_2 - 2t_1 = 4 + 5$

$\therefore t_2 - 2 = 9$

$\therefore \boxed{t_2 = 7}$

put $n = 2$ in eq^n (A) we get

$\therefore \quad 4c_1 + 8c_2 + c_3 + 2c_4 = 7 \quad —— (iii)$

put $n = 3$ in eq^n (1) we get

$t_3 - 2t_2 = 8 + 5$

$t_3 - 2 \times 7 = 13$

$\therefore \quad t_3 = 13 + 14 = 27 \qquad \therefore \boxed{t_3 = 27}$

put $n = 5$ in eq^n (A) we get

$$8c_1 + 24c_2 + c_3 + 5c_4 = 27 \quad \text{---- (iv)}$$

After solving eq^n (i), (ii), (iii), (iv) we get value of c_1, c_2, c_3, c_4

$$T(n) = \begin{cases} 10 & \text{if } n=0 \\ 4t(n-1) + 4^n(n+1) & \text{if } n \geq 1 \end{cases}$$

(S/18, 6)

$t_n - 4t_{n-1} = 4^n(n+1)$

$t_n - 4t_{n-1} = 4^n \cdot n + 4^n$ —— (A)

$a_0 t_n + a_1 t_{n-1} = b_1^n p_1(n) + b_2^n p_2(n)$

$a_0 = 1, \ a_1 = -4, \ k = 1$

$b_1 = 4, \ P_1(n) = n, \ d_1 = 1$

$b_2 = 4, \ P_2(n) = 1, \ d_2 = 0$

Characteristic eq^n is

$(a_0 x^k + a_1 x^{k-1})(x - b_1)^{d_1+1} (x - b_2)^{d_2+1}$

$(x-4)(x-4)^2(x-4)$

$r_1 = 4, \ r_2 = 4, \ r_3 = 4, \ r_4 = 4$

general eq^n is

$t_n = c_1 r_1^n + c_2 n r_2^n + c_3 n^2 r_3^n + c_4 n^3 r_4^n$ —— (A)

put $n = 0$ in A

$\boxed{c_1 = 10}$

put $n = 1$ in (A)

$t_1 - 4t_0 = 8$

$t_1 - 10 \times 4 = 8 \quad t_1 = 8 + 40 = 48$

Que:- Solve the following recurrance by method of characteristic equation: (S-18/7M)

$$t_n = \begin{cases} n & \text{if } n=0 \text{ or } n=1 \\ t_{n-1} + t_{n-2} & \text{if } n > 1 \end{cases}$$

(fibonacci series recurrence)

Ans:-

1) Rearrange the given expression

$t_n - t_{n-1} - t_{n-2} = 0$

2) Compare expression with

$a_0 t_n + a_1 t_{n-1} + a_2 t_{n-2} = 0$

$\therefore k = 2, \ a_0 = 1, \ a_1 = -1, \ a_2 = -1$

3) form the characteristic polynomial of degree k using constants as

$a_0 x^k + a_1 x^{k-1} + \cdots + a_k$

$\therefore$ Characteristic polynomial is

$x^2 - x - 1 = 0$

$\therefore$ The roots of the characteristic polynomial are

$r_1 = \frac{1+\sqrt{5}}{2}$ and $r_2 = \frac{1-\sqrt{5}}{2}$

[$\underset{a}{x^2} - \underset{b}{x} - \underset{c}{1} = 0$ $\qquad \frac{-b \pm \sqrt{b^2-4ac}}{2a}$

$\therefore$ General solution is of the form

$$\cancel{f_n =}\ t_n = c_1 r_1^n + c_2 r_2^n$$

$$\therefore t_n = c_1\left(\frac{1+\sqrt{5}}{2}\right)^n + c_2\left(\frac{1-\sqrt{5}}{2}\right)^n$$

$\therefore$ The initial conditions are

$\cancel{f}$ $\quad n = 0$ & $n = 1$

for $n=0$ we get

$$c_1 + c_2 = 0 \quad \text{———} (i)$$

for $n=1$ we get

$$\cancel{c_1\left(\frac{1+\sqrt{5}}{2}\right) + c_2\left(\frac{1-\sqrt{5}}{2}\right) = 1}$$

$$\cancel{\frac{1}{2}c_1 + \frac{\sqrt{5}}{2}c_1 + \frac{1}{2}c_2 - \frac{\sqrt{5}}{2}c_2 = 1}$$

$$c_1 r_1 + c_2 r_2 = 1 \quad \text{———} (ii)$$

from eq^n (i) we get

$$c_1 = -c_2$$

puting in eq^n (ii) we get

$$-c_2 r_1 + c_2 r_2 = 1$$

$$\therefore c_2(r_2 - r_1) = 1$$

$$\therefore c_2 = \frac{1}{r_2 - r_1}$$

$$= \frac{1}{\frac{1-\sqrt{5}}{2} - \frac{1+\sqrt{5}}{2}}$$

$$= \frac{1}{\cancel{\frac{1}{2}} - \frac{\sqrt{5}}{2} - \cancel{\frac{1}{2}} - \frac{\sqrt{5}}{2}}$$

$$= \frac{1}{-\frac{2\sqrt{5}}{2}}$$

$$= \frac{\cancel{2}}{-\cancel{2}\sqrt{5}} = -\frac{1}{\sqrt{5}}$$

$\therefore C_2 = -\frac{1}{\sqrt{5}}$ and $C_1 = \frac{1}{\sqrt{5}}$

$\therefore t_n = \frac{1}{\sqrt{5}} r_1^n - \frac{1}{\sqrt{5}} r_2^n$

$\therefore t_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$ — Ans.

w.

Que:- Solve the given recurrance (S-16/3M)

$$t_n = \begin{cases} 0 & \text{if } n=0 \\ 5 & \text{if } n=1 \\ 3t_{n-1} + 4t_{n-2} & \text{otherwise} \end{cases}$$

Que:- Solve the following eq^n by using characteristic eq^n (W-16 / 7M)

$$t_n = \begin{cases} n & \text{if } n=0,1,2 \\ 5t_{n-1} - 8t_{n-2} + 4t_{n-3} & \text{otherwise} \end{cases}$$

$n\log n \quad n^2 \quad n^3 \quad 2^n \quad n!$

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(n/2)+n & \text{if } n>1 \end{cases}$$

1) Solve the recurrence $T(n) = 2T(n/2)+n$ by substitution method (C7M)

Que:- Solve the following recurrance using substitution method by first guessing the solution and then use induction to find constants and show that the solution works (S-18/4M)

Soln:- 1. Guess the solution

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

$T(n) \leq cn\log n$

2. Now we use mathmatical induction

(i) Base case. Here our guess does not hold for $n=1$

because $T(1) \leq c.1.\log 1$

$T(1) \leq c\times 1\times \lg 1$ [$\because \log 1 = 0$]

$1 \leq c\times 1\times 0$

i.e $T(1) \leq 0$ which is contradiction with $T(1)=1$

Now for $n=2$

$T(2) \leq c.2\log 2$

$2T(2/2)+2 \leq c.2$

$2T(1)+2 \leq c.2$

$0+2 \leq c.2$

$2 \leq c.2$ which is True

$a=2,\ b=2,\ f(n)=n$

$n^{\log_b^a} = n^{\log_2^2} = n^1 = n$

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

$\therefore \Theta(n.\log n)$

$\therefore T(n) \leq c.n.\log n$ is true for $n=2$

Now we show that it is true for $n = n/2$

i.e $T(n/2) \leq c \cdot n/2 \log n/2$ is true

Now we show that it is true for $n = n$

i.e $T(n) \leq cn \lg n$

we know $T(n) \leq 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$

$$\leq 2\left(c\left\lfloor \frac{n}{2}\right\rfloor \log\left\lfloor \frac{n}{2}\right\rfloor\right) + n$$

$$\leq cn \log\left(\frac{n}{2}\right) + n \quad \leq cn\log n - cn \log 2$$

$$\leq cn \log n - cn + n \qquad [\because \log 2 = 1]$$

$$\leq cn \log n \ \forall c \geq 1$$

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

Scanned by CamScanner

$\sqrt{n}$	n	$n\log n$	n^2	n^3	2^n	$n!$

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

put ~~$n = \log m$~~ $m = \log n$

$\therefore$ ~~$m =$~~ $\therefore n = 2^m$

$\Rightarrow T(2^m) = T(2^{m/2}) + 1$

$\sqrt{n} = n^{1/2}$

$= (2^m)^{1/2}$

$= 2^{\frac{m}{2}}$

Now put valuee of 2^m $\quad 2^m = n$

$\therefore T(n) = T(n/2) + 1$

compare with

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$

$\therefore a = 1,\ b = 2,\ f(n) = 1$

$\therefore n^{\log_b a} = n^{\log_2 1} = n^0 = 1 \quad \therefore \Theta(1)$

$\therefore n^{\log_b a} = f(n)$

$\therefore$ second condition applies

$\therefore$ ~~$T(n) = \Theta(\log$~~

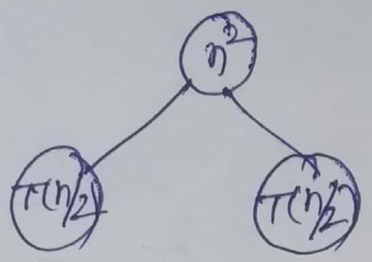
$T(n) = \Theta(n^{\log_b a} \cdot \log n)$

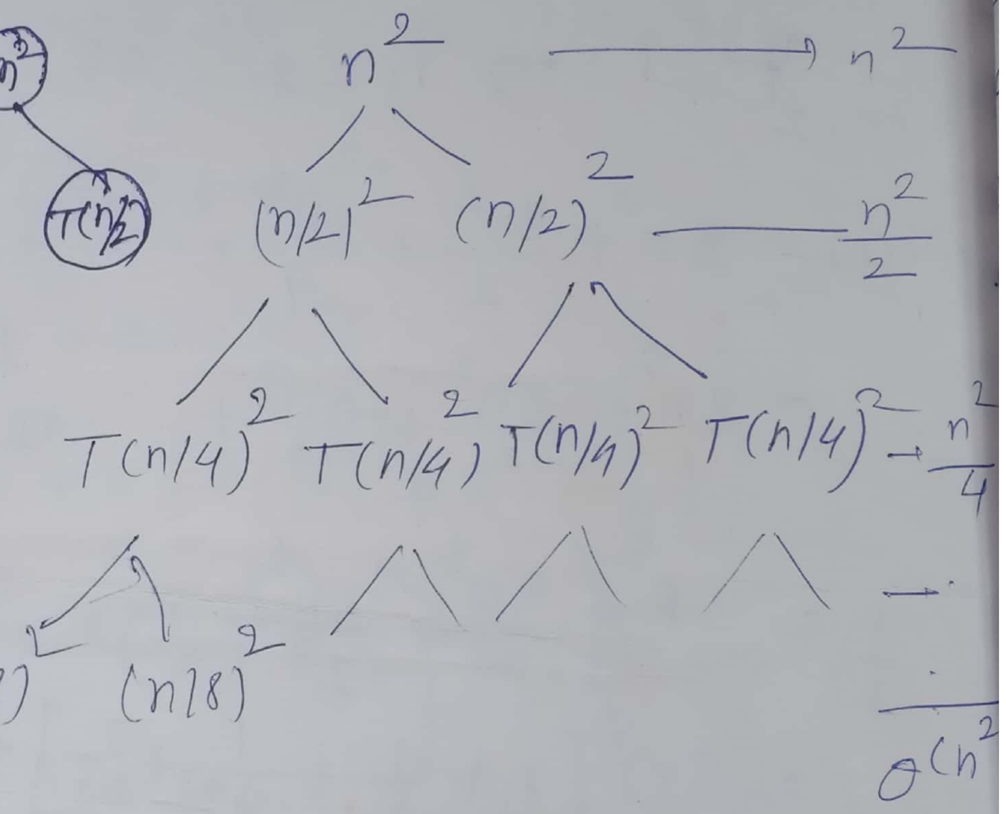
Recursion Tree

- Pictorial representation of an iteration method, [it] is in the form of a tree, where at each le[vel] nodes are expanded. In general, consider second [term] in recurrance as root. It is useful when divide & conquer [algorithm] is used.

eg: $T(n) = 2T\left(\frac{n}{2}\right) + n^2$

sol^n: Recursion tree is as follows





$\left(\frac{n}{2}\right)^2 + \left(\frac{n}{2}\right)^2$

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

$T(n) = n^2 + \frac{n^2}{4} + \frac{n^2}{4^2} + \ldots \lg n$ times

$\leq n^2 \sum_{i=0}^{\infty} \left(\frac{1}{4^i}\right)$

$\leq n^2 \left[\frac{1}{1-\frac{1}{4}}\right] \leq 4n^2$

$\therefore T(n) = \Theta(n^2)$

ster method

it is used for solving the following types of
currence

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ with $a \geq 1$ and $b > 1$

In this problem is devided into 'a' subproblems
each of size $\frac{n}{b}$ where a and b are positive
constants.

The cost of dividing the problem & combining the
results of the subproblem is described by the
function $f(n)$.

Master method depends on following theorms

Let $T(n)$ be defind on non-negative integers
by the recurrance
$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ where $a \geq 1$ & $b > 1$ be
constants & $f(n)$ be a function Then $T(n)$
can be bound asymptotically as follows

Theorm 1) If $f(n) = O(n^{\log_b^{a-\epsilon}})$ for some constant
$\epsilon > 0$, then

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

Theorm 2:- If $f(n) = \Theta(n^{\log_b^a})$ then

$$\boxed{T(n) = \Theta(n^{\log_b^a} \lg n)}$$

Theorm 3 :- If $f(n) = \Omega(n^{\log_b^a + \epsilon})$ for some constant $\epsilon > 0$ and if $af(\frac{n}{b}) \leq cf(n)$ for sum $c < 1$ and all sufficiently large n, then

$$\boxed{T(n) = \Theta(f(n))}$$

, $\boxed{af(\frac{n}{b}) \leq c f(n)}$ is called Regularity condⁿ

Que 1 :- Solve the following recurrance : (S-19/7) (W-17/7)

$T(n) = T(\frac{n}{4}) + \sqrt{n} + 4$ for $n >= 4$, $T(1) = 4$

Ans :- Given recurrance relation is

$$T(n) = T(\frac{n}{4}) + \sqrt{n} + 4$$

compare with equation

$$T(n) = aT(\frac{n}{b}) + f(n)$$

$\therefore a = 1$, $b = 4$, $f(n) = \sqrt{n} + 4$

$$n^{\log_b^a} = n^{\log_4^1} = n^0 = 1$$

$$\therefore f(n) > n^{\log_b^a}$$

$\therefore$ 3rd Condition is applied

Scanned by CamScanner

Check for regularity condition

$af\left(\frac{n}{b}\right) \leqslant c\, f(n)$

$a \times \frac{\sqrt{n}+4}{b} \leqslant c \times \sqrt{n}+4$

$\therefore \frac{1 \times \sqrt{n}+4}{4} \leqslant c \times \sqrt{n}+4$

put $n = 4$

$\therefore \frac{1 \times \sqrt{4}+4}{4} \leqslant c \times \sqrt{4}+4$

$\frac{1 \times 6}{4} \leqslant 6c$

$\therefore \frac{\cancel{1} \times \cancel{6}}{4 \times \cancel{6}} \leqslant c$

$\therefore c \geqslant \frac{1}{4}$

$\therefore \frac{1}{4} \leqslant c$ - regularity condition satisfi

$\therefore \boxed{T(n) = \theta\, f(n)}$

~~$T(n) = \theta(\sqrt{n})$~~ $\boxed{T(n) = \theta(\sqrt{n})}$ — Ans

Que:- Solve the following using master method

(CW/18-8)

1) $T(n) = 9T(n/3) + n$

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

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

4) $T(n) = 3T(n/4) + n \log n$

Ans:- 1) $T(n) = 9T(n/3) + n$

1) Compare given recurrance with

$T(n) = aT(n/b) + f(n)$

$\therefore a = 9, b = 3, f(n) = n$

$\therefore n^{\log_b^a} = n^{\log_3^9} = n^2$

as $f(n) < n^{\log_b^a}$,

first condition is applied

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

$\boxed{\therefore T(n) = \Theta(n^2)}$ — Ans

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

Compare given expression with

$T(n) = aT(\frac{n}{b}) + f(n)$

$\therefore a = 1, b = \frac{3}{2}, f(n) = 1$

$n^{\log_b^a} = n^{\log_{3/2}^1} = n^0 = 1$

$\therefore f(n) = n^{\log_b^a}$

$\therefore$ 2^{nd} condn satisfies

$\therefore T(n) = \theta(n^{\log_b^a} \lg n)$

$\boxed{T(n) = \theta(\lg n)}$ — Ans

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

Compare given expression with

$T(n) = aT(\frac{n}{b}) + f(n)$ we get

$\therefore a = 4, b = 2, f(n) = n$

$n^{\log_b^a} = n^{\log_2^4} = n^2$ $[\because \log_2^4 = 2]$

$\therefore f(n) < n^{\log_b^a}$

$\therefore$ first condition is applied

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

$\boxed{T(n) = \theta(n^2)}$ — Ans

4) $T(n) = 3T(\frac{n}{4}) + n\log n$

comparing given eqn with

$T(n) = aT(\frac{n}{b}) + f(n)$ we get

$a = 3, b = 4$ $f(n) = n\log n$

$\therefore n^{\log_b^a} = n^{\log_4^3} = n^{0.793}$

where $\epsilon = 0.2$

case 3 applies for regularity condition

$af(\frac{n}{b}) \leq c f(n)$

$3[\frac{n}{4}] \log[\frac{n}{4}] \leq \frac{3}{4} n \log n$

$\leq c f(n)$ for $c = \frac{3}{4}$

$\therefore T(n) = \Theta(f(n))$

$\boxed{T(n) = \Theta(n \log n)}$ – Ans.

Que:- Solve the following using master method (H.W) (S-17/8M)

1) $T(n) = 2T(\frac{n}{4}) + n$

2) $T(n) = 3T(\frac{8n}{4}) + n^2$

3) $T(n) = 6T(\frac{n}{8}) + \log n$

4) $T(n) = 7T(n/5) + \sqrt{n} + 2$

Ans:- 1) case 3 applies

$\boxed{\therefore T(n) = \Theta(f(n)) = \Theta(n)}$ – Ans.

2) case 3 applies

$\boxed{\therefore T(n) = \Theta(f(n)) = \Theta(n^2)}$ – Ans.

3) first condition applies (case 1)

$\therefore T(n) = \Theta(n^{\log_b^a}) = \Theta(n^{\log_8^6})$ – Ans

4) case 1 applies

$\therefore T(n) = \Theta(n^{\log_b^a}) = \Theta(n^{\log_5^7})$ — Ans.