SOLVED QUESTION BANK

[Sequence given as per syllabus]

INTRODUCTION

Mathematical induction is the most basic way to evaluate a series. It consists of basic and hypothesis phase. When an algorithm contains a recursive call to itself, its running time can often be described by a recurrence. This chapter offers three methods for solving recurrences, that is, for obtaining asymptotic 'θ' or 'O' bounds on the solution. These are substitution method, recursion tree method and master method. An algorithm is any well-defined computational procedure that takes an input and produces an output. This chapter is an overview of algorithms and their place in modern computing systems. This chapter defines what an algorithm is and lists some examples.

MATHEMATICAL FOUNDATION

Q.1. What is mathematical induction? Give its types.

Ans. Mathematical induction :

- Induction is a method to prove sequence of steps. It is the most basic way to evaluate a series.
- There are two phases of induction :

  (1) Basic phase

  (2) Induction/Hypothesis phase.
- If an equation is valid for n^{th} term then using induction, it is possible to prove the same equation valid for $(n+1)^{th}$ terms.

Example : Consider the following proof

Prove : $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Step I : Basic step :

Prove the induction for n = 1

$$L.H.S. = \sum_{i=1}^{1} i = 1$$

$$R.H.S. = \frac{n(n+1)}{2} = \frac{1(1+1)}{2} = \frac{2}{2} = 1$$

$\therefore$ L.H.S. = R.H.S.

Step II : Hypothesis phase :

Prove the induction for n = n + 1

$$L.H.S. = \sum_{i=1}^{n+1} i = 1 = \frac{(n+1)(n+2)}{2}$$

$$\sum_{i=1}^{n+1} i = \left[\sum_{i=1}^{n} i\right] + (n+1)$$

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

$$= \frac{n^2+3n+2}{2}$$

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

$\therefore$ L.H.S. = R.H.S.

Q.2. Prove : $\sum_{i=1}^{n} (2i-1) = n^2$.

Ans. Step I : Basic step :

Prove induction for n = 1

$$L.H.S. = \sum_{i=1}^{1} (2i-1) = 1$$

$$R.H.S. = n^2 = 1^2 = 1$$

$\therefore$ L.H.S. = R.H.S.

Step II : Hypothesis phase :

$$L.H.S. = \sum_{i=1}^{n+1} (2i-1) = (n+1)^2$$

$$\sum_{i=1}^{n+1} (2i-1) = \left[\sum_{i=1}^{n} (2i-1)\right] + 2(n+1) - 1$$

$$= \left[\sum_{i=1}^{n} (2i-1)\right] + (2n+1)$$

$$= n^2 + 2n + 1 \qquad \left[\because \sum_{i=1}^{n} (2i-1) = n^2\right]$$

$$= (n+1)^2$$

$\therefore$ L.H.S. = R.H.S.

SOLVED QUESTION BANK

[Sequence given as per syllabus]

INTRODUCTION

Mathematical induction is the most basic way to evaluate a series. It consists of basic and hypothesis phase. When an algorithm contains a recursive call to itself, its running time can often be described by a recurrence. This chapter offers three methods for solving recurrences, that is, for obtaining asymptotic 'θ' or 'O' bounds on the solution. These are substitution method, recursion tree method and master method. An algorithm is any well-defined computational procedure that takes an input and produces an output. This chapter is an overview of algorithms and their place in modern computing systems. This chapter defines what an algorithm is and lists some examples.

MATHEMATICAL FOUNDATION

Q.1. What is mathematical induction? Give its types.

Ans. Mathematical induction :

- Induction is a method to prove sequence of steps. It is the most basic way to evaluate a series.
- There are two phases of induction :

  (1) Basic phase

  (2) Induction/Hypothesis phase.
- If an equation is valid for n^{th} term then using induction, it is possible to prove the same equation valid for $(n+1)^{th}$ terms.

Example : Consider the following proof.

Prove : $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Step I : Basic step :

Prove the induction for n = 1

$$\text{L.H.S.} = \sum_{i=1}^{1} i = 1$$

$$\text{R.H.S.} = \frac{n(n+1)}{2} = \frac{1(1+1)}{2} = \frac{2}{2} = 1$$

∴ L.H.S. = R.H.S.

Step II : Hypothesis phase :

Prove the induction for n = n + 1

$$\text{L.H.S.} = \sum_{i=1}^{n+1} i = 1 = \frac{(n+1)(n+2)}{2}$$

$$\sum_{i=1}^{n+1} i = \left[\sum_{i=1}^{n} i\right] + (n+1)$$

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

$$= \frac{n^2+3n+2}{2}$$

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

∴ L.H.S. = R.H.S.

Q.2. Prove : $\sum_{i=1}^{n}(2i-1) = n^2$.

Ans. Step I : Basic step :

Prove induction for n = 1

$$\text{L.H.S.} = \sum_{i=1}^{1}(2i-1) = 1$$

$$\text{R.H.S.} = n^2 = 1^2 = 1$$

∴ L.H.S. = R.H.S.

Step II : Hypothesis phase :

$$\text{L.H.S.} = \sum_{i=1}^{n+1}(2i-1) = (n+1)^2$$

$$\sum_{i=1}^{n+1}(2i-1) = \left[\sum_{i=1}^{n}(2i-1)\right] + 2(n+1) - 1$$

$$= \left[\sum_{i=1}^{n}(2i-1)\right] + (2n+1)$$

$$= n^2 + 2n + 1 \qquad \left[\because \sum_{i=1}^{n}(2i-1) = n^2\right]$$

$$= (n+1)^2$$

∴ L.H.S. = R.H.S.

SUMMATION OF ARITHMETIC AND GEOMETRIC SERIES

Q.3. Explain summation of arithmetic series.

Ans. **Summation of arithmetic series :**

- An arithmetic series is the sum of a sequence $\{a_k\}$, $k = 1, 2, \ldots\ldots$, in which each term is computed from the previous one by adding or subtracting a constant d.

Therefore, for $k > 1$,

$$a_k = a_{k-1} + d = a_{k-2} + 2d = \ldots a_1 + d\,(k+1).$$

- The arithmetic series has the value

$$\sum_{k=1}^{n} k = \frac{1}{2} n(n+1) = \theta(n^2).$$

Q.4. Derive the closed form for the summation :

$$S_n = \sum_{k=1}^{n} a_k.$$

Ans. **Given :**

$$S_n = \sum_{k=1}^{n} a_k$$

$$= \sum_{k=1}^{n} [a_1 + (k-1)d]$$

$$= n\,a_1 + d \sum_{k=1}^{n} (k-1)$$

$$= n\,a_1 + d \sum_{k=1}^{n-1} k$$

Using the sum identity

$\sum_{k=1}^{n} k = \frac{1}{2} n(n+1)$, we get

$$S_n = na_1 + \frac{1}{2} dn(n-1) = \frac{1}{2} n[2a_1 + d(n-1)]$$

However,

$$a_1 + a_n = a_1 + [a_1 + d(n-1)$$

$$= 2a_1 + d(n-1)$$

So,

$$S_n = \frac{1}{2} n(a_1 + a_n)$$

Q.5. Derive a closed form solution for the summation:

$$\sum_{i=1}^{k-1} i\,a^i$$

CT : W-12(7M)

Ans. Assume $S_k = \sum_{i=1}^{k-1} i\,a^i$

$$\therefore\ t_2 = 3t_1 + 2 + 4 \qquad \ldots(1)$$

We have

$$a S_k = a^2 + 2a^3 + 3a^4 + \ldots + (k-1)a^k \qquad \ldots(2)$$

Subtracting the second equation from the first, we obtain

$$(1-a)\ S_k = a + a^2 + a^3 + \ldots + a^{k-1} + a^k - ka^k$$

$$= a(1 + a + a^2 + \ldots + a^{k-1}) - ka^k$$

$$(1-a)\ S_k = a(1-a^k)/(1-a) - ka^k$$

[∵ sum of first k terms of G.P.]

$$\therefore\ S_k = \frac{a(1-a^k)/(1-a)}{(1-a)} - ka^k$$

$$\therefore\ S_k = \sum_{i=1}^{k-1} i\,a^i = \frac{a(1-a^k)}{(1-a)^2} - \frac{ka^k}{(1-a)}$$

Q.6. Explain the geometric series of summation.

Ans. **Geometric series :**

- Geometric series is the ratio of successive term.
- In geometric series the ratio of successive terms are constant, so that the series may be represented as a, ar, ar^2, ... ar^{n-1}
- Geometric series has the value $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$

Q.7. Prove : $\sum_{i=0}^{n} a_r{}^i = \frac{a(r^{n+1}-1)}{r-1}$ for all n >= 0.

Ans. If initial value of n is given then basic must start with initial value otherwise we use basic n = 1.

Step I : Basic step :

For n = 0

$$\text{L.H.S.} = \sum_{i=0}^{0} ar^i = ar^0 = a$$

$$\text{R.H.S.} = \frac{a(r^1 - 1)}{(r-1)} = a$$

$\therefore$ L.H.S. = R.H.S.

Step II : Hypothesis phase :

For n = n + 1

$$\text{R.H.S.} = \frac{a(r^{(n+2)} - 1)}{r-1}$$

$$\text{L.H.S.} = \sum_{i=0}^{n+1} ar^i = \left[\sum_{i=0}^{n} ar^i\right] + a\,r^{n+1}$$

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

$$= \frac{a\,r^{n+1} - a + (r-1)\,ar^{n+1}}{r-1}$$

$$= \frac{a\,r^{n+1} - a + ar^1\,ar^{n+1} - ar^{n+1}}{r-1}$$

$$\therefore \text{L.H.S} = \frac{a(r^{n+2} - 1)}{r-1}$$

$\therefore$ L.H.S = R.H.S.

BOUNDING SUMMATION USING INTEGRATION

Q.8. Give the techniques of bounding summation.

Ans. **Bounding summation** : There are many techniques available for bounding the summation that describe the running times of algorithm. They are :

(i) Mathematical induction.

(ii) Bounding the terms.

(iii) Splitting summations.

(iv) Approximation by integrals.

Q.9. Explain the concept of summation using integration to find lower bound and upper bound.

CT : W-06(4M), S-13(5M)

Ans. **Bounding summation using integration :**

- When a summation can be expressed as $\sum_{k=m}^{n} f(k)$ Where, f (k) is monotonically increasing function, we can approximate it by integrals :

$$\int_{m-1}^{n} f(x)\,dx \le \sum_{k=m}^{n} f(k) \le \int_{m}^{n+1} f(x)\,dx\,.$$

- When f (k) is monotonically decreasing function we can use a similar method to provide the bounds.

$$\int_{m}^{n+1} f(x)\,dx \le \sum_{k=m}^{n} f(k) \le \int_{m-1}^{n} f(x)\,dx\,.$$

- This integral approximation gives a tight estimate for the n^{th} harmonic number. For a lower bound, we obtain.

$$\sum_{k=1}^{n} 1/k \ge \int_{1}^{n+1} \frac{dx}{x}$$

$$= \ln (n+1)$$

- For the upper bound, we derive the inequality

$$\sum_{k=2}^{n} 1/k \le \int_{1}^{n} \frac{dx}{x}$$

$$= \ln n$$

which yields the bound

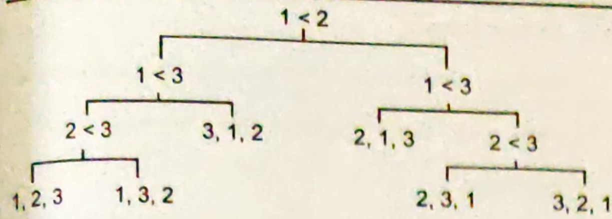
$$\sum_{k=1}^{n} 1/k \le \ln n + 1$$

Q.10. Show the lower bound for any sorting algorithm which does sorting by comparisons of keys is n log n.

CT : S-06, W-12(4M), W-08(7M), W-10, S-13(6M)

Ans.

- The lower bound for most sorting algorithm is $\Omega(n \log n)$. This is because most sorting algorithm use item comparisons to determine the relative order of items.
- Any algorithm that sorts by comparisons will have a minimum lower bound of $\Omega(n \log n)$ because a comparison tree is used to select a permutation that is sorted.
- A comparison tree for three numbers 1, 2 and 3 can be easily constructed.



- Every item is compared with every other item, and that each path results in valid permutation of the three items.
- The height of the tree determines the lower bound of the sorting algorithm. Because there must be as many leaves as there are permutations for the algorithm to be correct, the smallest possible height of the comparison tree is log N ! which is equivalent to Ω (N log N) .

Q.11. Show that log n ! = O(n log n). **CS : S-14(3M)**

OR Prove that :

$$\sum_{i=1}^{n} \log (i) = \theta (n \log n)$$ **CS : S-14(3M)**

Ans. Method 1 :

By stririllings approximation :

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \theta\left(\frac{1}{n}\right)\right)$$

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

$$\log n! = \log\left(\frac{n}{e}\right)^n$$

$$= n \log \frac{n}{e} \qquad [\because \log a^b = b \log a]$$

$$= n \log n - n \log e$$

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

Method 2 :

log n ! = O (n log n) can be solved by another method as follows :

We know that n ! = n (n – 1) (n – 2) 3. 2. 1

$$\Rightarrow n! \geq n(n-1)(n-2).....\left(\left\lceil\frac{n}{2}\right\rceil\right)$$

$$\Rightarrow n! \geq \left(\frac{n}{2}\right)^{\frac{n}{2}} \qquad ....(1)$$

Taking log on both sides of equation (1)

$$\log n! \geq \log\left(\frac{n}{2}\right)^{\frac{n}{2}}$$

$$\log n! \geq \frac{n}{2}\log\frac{n}{2}$$

$$\log n! \geq \frac{n}{2}(\log n - \log_2^2)$$

$$\log n! \geq \frac{n}{2}(\log n - 1) \Rightarrow \log n! \geq n \log n \qquad [\because \log_a a = 1]$$

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

Q.12. Derive lower bound for $\sum_{i=1}^{n-1} 1/i^2$ using the concept of summation using integration. **CT : W-06(3M)**

Ans. **Given :** $\sum_{i=1}^{n-1} 1/i^2$

$$= \sum_{i=1}^{n} \log i \qquad \left[\because \sum_{i=1}^{n} 1/i^2 = \log(n)\right]$$

$$= \sum_{i=1}^{n} \log n$$

using integration $\int \log n \, dn$

first function = log n

second function = 1

$$\int I.II = \log n \int 1.dn - \int \frac{1}{n} \int 1.dn$$

$$= n \log n - \int \frac{1}{n} n \, dn$$

$$\sum_{i=1}^{n-1} 1/i^2 = n \log n - n$$

RECURSION AND INDUCTION

Q.13. What is recurrence relation?

Ans. Recurrence relation :

- A recurrence relation is an equation which is defined in terms of itself.

(i) Many natural functions are easily expressed as recurrences :

$a_n = a_{n-1} + 1, a_1 = 1 \rightarrow a_n = n$ (Polynomial)

$a_n = 2a_{n-1}, a_1 = 1 \rightarrow a_n = 2^{n-1}$ (Exponential)

(ii) It is often easy to find a recurrence as the solution of a counting problem.

Q.14. Explain recursion in mathematical induction.

Ans. Recursion in mathematical induction :

- In both, we have general and boundary conditions, with the general condition breaking the problem into smaller and smaller pieces.
- The initial and boundary condition terminate the recursion.

Example (1) : $T_n = 2T_{n-1} + 1,\ T_0 = 0$

n	0	1	2	3	4	5	6	7
T_n	0	1	3	7	15	31	63	127

Example (2) : Prove that $T_n = 2^n - 1$ by induction.

Steps :

(1) Show that the basis is true :

$T_0 = 2^0 - 1 = 1 - 1 = 0$

(2) Now assume it is true for $(n-1)$ i.e. T_{n-1}.

(3) Using this assumption show for T_n :

$\therefore\ T_n = 2T_{n-1} + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 1$

- Thus one can define recurrence as :

It is a equation or inequality that defines function in terms of its value or smaller inputs.

Q.15. What are the different methods of solving recurrence? Explain Master method in detail.

CT : S-06,11(8M), S-08(7M), CS : W-11(2M)

Ans. There are five methods of solving recurrence :

(i) Substitution method.

(ii) Iteration method.

(iii) Recursion tree.

(iv) Changing variables.

(v) Master theorem.

(i) **Substitution method :**

In this method "To guess the solution, play around with small values for insight" is in this method first one start guess a solution and prove by induction.

(ii) **Iteration method :**

- This method is also known as "Try back substituting until we known what is going". That is "plug the recurrence back into itself until we see a pattern".
- In this method, recurrence is expanded as summation of terms, then summation provides the solutions.

(iii) **Recursion tree :**

- Recursion tree is defined as "Drawing a picture of the back substitution process gives us an idea of what is going on".
- In recursion tree we must keep track of two things:

(a) Size of the remaining argument to the recurrence.

(b) The additive stuff to be accumulated during this call.

(iv) **Changing variables :**

In change variable method as the name implies we change any complex term present in recurrence by any variable.

Working procedure :

(1) Select a variable in place of complex term and put the variable as a substitute in the recurrence equation.

(2) For finding actual result replace the choosen variable by the complex term.

(v) **Master theorem :**

- Let $a \geq 1$ and $b \geq 1$ be constants, let f (n) be a function and let T(n) be defined on the non negative integers by the recurrence.

$$\boxed{T(n) = a\,T\left(\frac{n}{b}\right) + f(n)}$$

Where, we interpret $\frac{n}{b}$ to mean either $\left\lfloor \frac{n}{b} \right\rfloor$ or $\left\lceil \frac{n}{b} \right\rceil$.

- Then T(n) can be bounded asymptotically as follows :

(1) If $f(n) = \theta(n \log_b^a - \epsilon)$ for some constant $\epsilon > 0$ then

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

(2) If $f(n) = \theta(n \log_b^a)$, then

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

(3) If $f(n) = \Omega(n \log_b^a + \epsilon)$ for some constant $\epsilon > 0$ and if $a f\left(\frac{n}{b}\right) \le c f(n)$ for some constant $c < 1$ and all sufficiently large n, then

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

- Here, one can see that f (n) is compared with $n \log_b^a$ and solution to recurrence obtained by selecting maximum of two as in case 1 and case 3. In case 2 the two function are of the same size, so we multiply by logarithmic function thus,

$T(n) = \theta(n \log_b^a \log(n))$

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

Q.16. Explain the difference between substitution and recursive tree method used for solving recurrence problems. ***CT : S-14(4M)***

Ans. **Substitution method :**

- It involves guessing the form of the solution and then using mathematical induction to find the constants and show that the solution works.
- This method is powerful but it can be applied only in cases when it is easy to guess the form of the answer.
- The substitution method can be used to establish either upper or lower bounds on a recurrence.

Recursion Tree :

- Recursion tree method is a pictorial represent-tation of an iteration method which is in the form of a tree, where at each level nodes are expanded.
- In general case we consider second term in recurrence as a root.
- It is useful when divide and conquer algorithm is used.

Q.17. Solve the following recurrence relation.

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

and $T(1) = 1$. ***CT : S-06(5M)***

OR **Solve the recurrence relation**

$t(n) = t(n-1) + n$ if $n \ge 2$

$t_n = 1$ if $n = 1$. ***CT : S-09(7M)***

Ans. **Step I :**

Generate recurrence :

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

Step II :

Consider L.H.S :

$t_n - t_{n-1}$

Root is $(x - 1)$

$\therefore r_1 = 1$

Consider R.H.S.

$n \Rightarrow b^n . P(n)$

Here, $b = 1$ $P(n) = n$. Root is always $(x - b)$

As $P(n) = n$, degree = 1

$\therefore$ Root = twice (d + 1)

$(x-1)(x-1)$, $r_2 = r_3 = 1$

Step III :

Roots are real and same :

$\therefore t_n = C_1 r_1^n + C_2 . n . r_2^n + C_3 n^2 r_3^n$ ....(1)

Step IV :

Using given recurrence :

$t(1) = 1$ at $n = 1$

At n = 1 in equation (1),

$t(1) = C_1 + C_2 + C_3$

$C_1 + C_2 + C_3 = 1$ ......(A)

Using given recurrence, at n = 1

$T(1) = T(0) + 1 = 1$

At n = 2 in equation (1),

$t_2 = C_1 + 2C_2 + 4C_3$

Using recurrence, at n = 2

$t(2) = t(1) + 2 = 1 + 2 = 3$

$\therefore C_1 + 2C_2 + 4C_3 = 3$ ......(B)

At n = 3 in equation (1),

$t_3 = C_1 + 3C_2 + 9C_3$

$C_1 + 3C_2 + 9C_3 = t_3$

Using recurrence at n = 3

$t_3 = t(2) + 3 = 3 + 3$

$t_3 = 6$

$\therefore C_1 + 3C_2 + 9C_3 = 6$ ......(C)

Solving equation A, B, and C we get

$C_1 = 0, C_2 = \frac{1}{2}, C_3 = \frac{1}{2}$

Standard equation = $t_n = \frac{1}{2} n (1)^n + \frac{1}{2} n^2 (1)^n$

Q.18. Solve the recurrence relation

$T(n) = 9T(n/3) + n^2$ otherwise

and $T(n) = 1$ for $n = 1$.

CT : W-06(4M)

Ans. Step I :

Let $n = 3^i, i = \log_3 n, n^2 = 9^i$

Then recurrence can be written as :

$t_n = 9T(3^i/3) + 3^{i^2}$ *[$\because (3^i)^2 = (3^2)^i = 9^i$]*

$= 9T^{(3^{i-1})} + 9^i$

$t_n - 9T^{(3^{i-1})} = 9^i$ ....(1)

Step II :

$L.H.S. = t_n - 9T^{(3^{i-1})}$

Root is $(x-9)$ i.e. $r_1 = 9$

R.H.S. $9^i \Rightarrow b^i . P(i)$

Here, $b = 9$, $P(i) = 1$, $d = 0$, Root = once.

$\therefore (x-9)$

$\therefore r_2 = 9$

Step III :

Roots are real and same :

$\therefore t(i) = C_1 r_1^i + C_2 . i . r_2^i$

$n = 3^i, i = \log_3 n ; n^2 = 9^i$

$t(n) = C_1 9^i + C_2 . \log_3 n . 9^i$

$t(n) = C_1 n^2 + C_2 \log_3 n . n^2$ ....(A)

$T(n) = 1$ for $n = 1$ (given)

$T(1) = C_1 + 0$

$T(1) = C_1$

i.e. $C_1 = 1$

As $C_1 = 1$

and C_2 is not solvable.

$\therefore Tn = n^2 + C_2 \log_3 n \; n^2$

Q.19. Solve the following recurrence relation.

CT : W-06(4M)

$T(n) = 2T(n/2) + n \log n$ otherwise

and $T(n)$ 1 for $n = 1$

Ans. Step I :

Let $n = 2^i$

$\therefore i = \log_2 n$

Then recurrence can be written as :

$t_n = 2t(2^i/2) + 2^i . i$

$\therefore t_n = 2t(2^{i-1}) + 2^i . i$

$\therefore t_n - 2t(2^{i-1}) = 2^i . i$ ....(1)

Step II :

$L.H.S. = t_n - 2t(2^{i-1})$

Root is $(x-2) = 0$

$\therefore r_1 = 2$

$R.H.S. = 2^i . i \Rightarrow b^i . P(i)$

$\therefore b^i = 2^i \Rightarrow b = 2 ; P(i) = i$

$\therefore d = 1$

Root twice

$(x-2)(x-2)$

$\therefore r_2 = 2, r_3 = 2$

Step III :

Standard solution :

Roots are real and same

$\therefore t(i) = C_1 r_1^i + C_2 . i . r_2^i + C_3 . i^2 . r_3^i$

Change base $n = 2^i, i = \log_2 n$

$\therefore t(n) = C_1 . n + C_2 . n . \log_2 n + C_3 . n . (\log_2 n)^2$

$T(n) = 1$ for $n = 1$

$\therefore T(1) = C_1$

$\therefore C_1 = 1$

as C_2 and C_3 is unsolvable

$\therefore t(n) = n + C_2 \; n . \log_2 n + C_3 . n (\log_2 n)^2$

Q.20. Solve the following recurrence relation :

$$T(n) = \begin{cases} 0 & \text{if } n = 0 \\ 3T(n/2) + n & \text{otherwise} \end{cases}$$

using method of substitution. Assume n is a power of 2.

CT : S-07, W-09(7M)

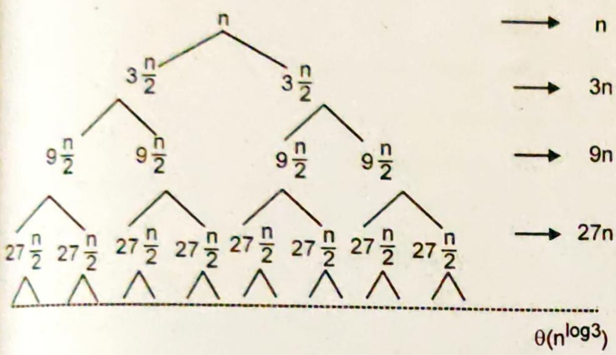
OR **Solve following logarithmic recurrence**

$$t_n = \begin{cases} 1 & \text{if } n = 1 \\ 3T(n/2) + n & \text{if n is power of 2 } (n > 1) \end{cases}$$

CS : S-12(5M)

Ans. Given that,

$$T(n) = 3T\left(\frac{n}{2}\right) + n$$



$$\Rightarrow T(n) = \theta(n \log_2{}^3) \text{ as } \log_2{}^3 > 1$$

Verification by substitution method :

$$T(n) = 3T\left(\frac{n}{2}\right) + n$$

Now, we compute first few values as power of 2.

n	1	2	4	8	16	32
T(n)	1	5	19	65	211	655

Now, we guess the solution as

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

n	T(n)
1	1
2	$3 \times 1 + 2$
4	$3^2 \times 1 + 3 \times 2 + 2^2$
8	$3^3 \times 1 + 3^2 \times 2 + 2^2 \times 3 + 2^3$
16	$3^4 \times 1 + 3^3 \times 2 + 3^2 \times 2^2 + 3 \times 2^3 + 2^4$

This shows that at level 1 of recursion tree number of child is one of level 2 number of child is two which multiple of 3, at level 2 number of child is C_1 which is also multiple of 3 and so on. This is shown in recursion tree.

Now, for n = 0

$$T(0) = 3^0 - 2^0 = 1 - 1 = 0$$

Let us assume that k > 0

$$\Rightarrow T(2^k) = 3^k 2^0 + 3^{k-1} 2^1 + 3^{k-2} 2^2$$

$$+ \ldots + 3.2^{k-1} + 3^0.2^k$$

$$= 3k \sum_{i=0}^{k} \left(\frac{2}{3}\right)^i$$

$$= 3^{k+1} - 2^{k+1}$$

$$\therefore \quad T(n) = 3^{n+1} - 2^{n+1}$$

Q.21. Give recursive algorithm to find the sum of first 'n' numbers of Fibonacci series obtain its recurrence relation also.

CT : S-07(7M)

Ans. Algorithm rec - fibo(n)

```
{
    if (n = 0) or (n = 1) then
        return (n)
    else
        return (rec-fibo (n –1)+ rec-fibo (n –2)
}
```

Complexity = x (n – 1) + x (n – 2)

Recurrence relation :

$$f(n) = \begin{cases} n & \text{if } n = 0 \text{ or } n = 1 \\ f(n-1) + f(n-2) & \text{otherwise} \end{cases}$$

$$t(n) = \begin{cases} n & \text{if } n = 0 \text{ or } n = 1 \\ t_{n-1} + t_{n-2} & \text{otherwise} \end{cases}$$

Step I :

Generate the characteristics equation :

$$\begin{array}{cccc} t_n & -t_{n-1} & -t_{n-2} & 0 \\ \downarrow & \downarrow & \downarrow & \\ 2 & 1 & 0 & \end{array}$$

Find degree of characteristics equation.

Step II :

Find generator polynomial :

$$x^2 - x - 1 = 0$$

Step III :

Find roots of generator polynomial :

$$r_1 = \frac{1+\sqrt{5}}{2},$$

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

Roots are real and distinct.

Applying standard solution

$$t_n = C_1 r_1^n + C_2 r_2^n + \ldots\ldots + C_n r_n^n$$

Step IV :

Using the initial condition solving recurrence,

$$t_n = C_1 . r_1^n + C_2 r_2^n \quad \ldots\text{(A)}$$

At $n = 0$, $t_n = 0$

$$\therefore C_1 + C_2 = 0 \quad \ldots(1)$$

At $n = 1$, $t_n = 1$

$$\therefore C_1\left[\frac{1+\sqrt{5}}{2}\right] + C_2\left[\frac{1-\sqrt{5}}{2}\right] = 1 \quad \ldots(2)$$

Using equation (1)

$$C_1 = -C_2$$

$$\Rightarrow C_1\left[\frac{1+\sqrt{5}}{2}\right] - C_1\left[\frac{1-\sqrt{5}}{2}\right] = 1$$

$$\Rightarrow \frac{C_1}{2} + \frac{C_1\sqrt{5}}{2} - \frac{C_1}{2} + \frac{C_1\sqrt{5}}{2} = 1$$

$$\Rightarrow 2\frac{C_1\sqrt{5}}{2} = 1$$

$$\therefore C_1 = \frac{1}{\sqrt{5}}$$

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

Replace C_1, C_2, r_1, r_2 in equation (A)

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

Q.22. Solve the given recurrence relation.

$t_n - 2t_{n-1} = (n+5)3^n$ for $n \geq 1$. **CT : W-07,S-11(7M)**

Ans. Step I :

L.H.S. $= t_n - 2t_{n-1}$

Root is $(x-2) = 0$

$$\therefore r_1 = 2$$

R.H.S. $= b^n . P(n) = (n+5)3^n$

$$\therefore b^n = 3^n$$

$P(n) = n + 5$

↑ ↑

1 0

degree = 1

$\therefore$ Root twice

i.e. $(x - b)$

$(x-3)(x-3)$

$\therefore r_2 = 3, r_3 = 3.$

Step II :

Roots are real distinct and same.

$$\therefore t_n = C_1 r_1^n + C_2 r_2^n + C_3 . n . r_3^n$$

$$\therefore t_n = C_1 2^n + C_2 3^n + C_3 . n . 3^n \quad \ldots(1)$$

Step III :

Using the given recurrence

$$t_n = (n+5) 3^n + 2 t_{n-1} \quad \ldots(2)$$

At $n = 0$ using equation (1)

$$t_0 = C_1 + C_2 \quad \ldots(3)$$

At $n = 1$ using equation (2)

$$t_1 = 2 t_0 + 18$$

At $n = 1$ using equation (1)

$$\therefore 2C_1 + 3C_2 + 3C_3 = 2 t_0 + 18 \quad \ldots\text{(A)}$$

At $n = 2$ using equation (2)

$$t_2 = 63 + 2t_1 \Rightarrow 63 + 2(2 t_0 + 18)$$

$$= 4 t_0 + 99$$

At $n = 2$ using equation (1)

$$t_2 = 4C_1 + 9C_2 + 18C_3$$

$$4C_1 + 9C_2 + 18C_3 = 4 t_0 + 99 \quad \ldots\text{(B)}$$

Replace t_0 in equation (A) and (B), we get

(A) $\Rightarrow 2C_1 + 3C_2 + 3C_3 = 2(C_1 + C_2) + 18$

$$\Rightarrow 2C_1 + 3C_2 + 3C_3 = 2C_1 + 2C_2 + 18$$

$$\therefore C_2 + 3C_3 = 18 \quad \ldots(4)$$

(B) $\Rightarrow 4C_1 + 9C_2 + 18C_3 = 4(C_1 + C_2) + 99$

$$\Rightarrow 4C_1 + 9C_2 + 18C_3 = 4C_1 + 4C_2 + 99$$

Solving equation (4) and (5), we get

$C_3 = 3, C_2 = 9$

C_1 is not solvable

Final solution :

$$\therefore t_n = C_1 2^n + 9.3^n + 3.n.3^n$$

[$\because C_1$ is not solvable]

3. Solve the recurrence relation

$a_n = 3a_{n-1} - 3a_{n-2} + 3a_{n-3}$ where $a_0 = 0$

$a_3 = 3, a_5 = 10$. **CT : S-07(7M)**

$$t(n) = \begin{cases} n & \text{if } n = 0, 3, 5 \\ 3t_{n-1} - 3t_{n-2} + 3t_{n-3} \end{cases}$$

$$\begin{array}{cccc} t_n & -3t_{n-1} & +3t_{n-2} & -3t_{n-3} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 2 & 1 & 0 \end{array}$$

Quadratic equation :

$x^3 - 3x^2 + 3x - 3 = 0$

$x_1 = 2.25$

$x_2 = 0.37$

$x_3 = 0.37$

$\therefore$ Roots are $r_1 = 2.25,\ r_2 = 0.37,\ r_3 = 0.37$

Standard solution:

$c_1 r_1^n + c_2 r_2^n + c_3 \cdot n \cdot r_3^n$

Replace r_1, r_2, r_3 in above equation.

$\therefore c_1 \cdot (2.25)^n + c_2 \cdot (0.37)^n + c_3 \cdot n \cdot (0.37)^n$

$t(n) = (2.25)^n \cdot c_1 + (0.37)^n \cdot c_2 + c_3 \cdot n \cdot (0.37)^n$

use constant values i.e.

$$t_n = \begin{cases} 0 & \text{if } n = 0 \\ 3 & \text{if } n = 3 \\ 10 & \text{if } n = 5 \end{cases}$$

Put n = 0

$\therefore t(n) = (2.25)^0 \cdot c_1 + (0.37)^0 \cdot c_2 + c_3 \cdot 0 \cdot (0.37)^0$

$= 1.\ c_1 + c_2 + 0$

$\therefore c_1 + c_2 = 0$

$c_1 = -c_2$

Put n = 3

$\therefore t(n) = (2.25)^3 \cdot c_1 + (0.37)^3 c_2 + c_3 \cdot 3 \times (0.37)^3$

$\therefore 3 = 11.39\ c_1 + 0.050\ c_2 + 0.15\ c_3$

Put n = 5

$\therefore t(n) = (2.25)^5 \cdot c_1 + (0.37)^5 c_2 + c_3 \cdot 5 \times (0.37)^5$

On solving, we get

$\therefore 0 = 57.66\ c_1 + 6.93 \times 10^{-3}\ c_2 + 34.67 \times 10^{-3}\ c_3$

$c_2 = -9.383 \times 10^{-3}$

$c_3 = 5.13 \times 10^{-3}$

Now, $c_1 = -c_2$

$\therefore c_1 = 9.383 \times 10^{-3}$

$c_2 = -9.383 \times 10^{-3}$

$c_3 = 5.13 \times 10^{-3}$

$\therefore t(n) = (2.25)^5 \cdot c_1 + (0.37)^5 c_2 + c_3 \cdot 5 \times (0.37)^5$

$10 = 57.66\ c_1 + 6.93 \times 10^{-3}\ c_2 + 0.034\ c_3$ ....(3)

On solving equation (1) (2) and (3), we get

$c_1 = 0.169$

$c_2 = -0.169$

$c_3 = 7.20$

$t_n = (2.25)^n \times 0.169 + (0.37)^n \times (-0.169)$

$+ 7.20 \times n \times (0.37)^n$

Q.24. Give recursive algorithm to calculate, com-pound interest obtain its recurrence relation. **CT : W-07(7M)**

Ans. Algorithm compound interest (C, SI, Pr, rate, noy) ;

```
{
    SI = (Pr × rate × noy) / 100 ;
        if (C = "Y") then
            return SI ;
            else
            return 0 ;
}
```

SI	Frequency	Total steps
0	-	-
0	-	0
1	1	1
1	n	n
1	1	1
-	-	-
1	1	1
0	0	0
		i.e. n + 3

Recurrence relation = n + 3

VBD

Q.25. Obtain recurrence relation for sum of series

S = 1 + 2 + 3 + ... + n.

Solve this recurrence relation using method of substitution.

CT : W-07(6M)

Ans. Recurrence relation for sum of series :

S = 1 + 2 + 3 + ... +n.

$$t_n = \begin{cases} n \text{ if } n <= 0 \\ tn\,(n-1) + 1 \end{cases}$$

Step 1 :

Construct the recurrence using variable part.

$t_n \quad -t_n(n-1) = 0$

$\downarrow \quad \downarrow$

1 0

Step 2 :

Construct generating polynomial by finding degree of recurrence degree = 1

$\therefore (x-1) = 0$

Step 3 :

Finding roots

$r_1 = 1$

R.H.S. = 1

$= b^n \cdot P(x)$

$\therefore$ b = 1, P(x) = 1, d = 0, Root = once.

$\therefore$ Root = (x – b) = (x – 1)

$\therefore r_2 = 1$

Step 4 :

Roots are real and same

$t_n = C_1 r_1^n + C_2 \cdot n \cdot r_2^n$

$= C_1 1^n + C_2 \cdot n \cdot 1^n$

Put n = 0

$t_0 = C_1 + 0$

$t_0 = 0$

$C_1 = 0$

C_2 is not solvable.

Q.26. Derive the general solution of the recurrence relation :

T(n) = a T (n/b) + f (n)

T (1) = 1.

CT : W-08, S-11,12(6M)

Ans. T (n) = a T (n/b) + f (n)

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

$$= f(n) + a\,T\left(\frac{n}{b}\right) + a^2\,T\left(\frac{n}{b^2}\right)$$

$$= f(n) + a\,f\left(\frac{n}{b}\right) + a^2\,f\left(\frac{n}{b^2}\right) + ...a^i\,T\left(\frac{n}{b^i}\right)$$

$$\left[\because \frac{n}{b^i} = 1 \Rightarrow n = b^i \Rightarrow i = \log_b^n\right]$$

$$\therefore T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + ...a^{\log_b^n}\,T(1)$$

$$= f(n) + a\,f\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right)$$

$$+ .....a^{\log_b^{n-1}}\,T\left(\frac{n}{B^{\log_b^{n-1}}}\right) + \theta\left(n \log_b^a\right)$$

$$\left[\because a^{\log_b^n} = n^{\log_b^a}\right]$$

$$\therefore T(n) = \sum_{j=0}^{\log_b^{n-1}} a^j\,f\left(\frac{n}{b^j}\right) + \theta\left(n \log_b^a\right)$$

Hence proved.

Q.27. Solve the following recurrence relation :

(i) T(n) = 7 T(n/2) + 18 n² T(1) = 1

(ii) T(n) = 2T + $\log_2^n$.

CT : W-08(7M)

Ans.

(i) T(n) = 7 T(n/2) + 18 n² T(1) = 1 :

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

$\Rightarrow a = 7, b = 2, f(n) = 18n^2$

Now, we find $n \log_b^a$

$\Rightarrow n \log_2^7 = n^{2.81}$

$O(n \log_2^7 - \epsilon) = O(n^{2.81} - 0.81)$

$7(n) = O(n^2)$

The master theorem is satisfied

$\Rightarrow T(n) = \theta(n \log_b^a)$

$T(n) = \theta(n^{2.81})$

(ii) $T(n) = 2T + \log_2^n$:

The above problem is not correct.

Q.28. Show that the solution to

$T(n) = 2T([n/2]+7) + n, n \geq 2$ is O (n log n). **CT : W-11(7M)**

Ans. **Given :**

$T(n) = 2T([n/2]+7) + n$ ....(1)

Let us guess that $T(n) \in O(n \log n)$ i.e. we guess that the solution is O (n log n)

we have to prove that

$T(n) \leq (C_n \log n)$ for an appropriate choice of $C > 0$

We start by assuming that this bound holds for

$\left[-\frac{n}{2}\right]$ i.e.

$T\left(\left[\frac{n}{2}\right]\right) \leq C\left\lfloor\frac{n}{2}\right\rfloor \log\left(\left\lfloor\frac{n}{2}\right\rfloor\right)$

Substituting into the recurrence yields

$T(n) \leq 2(C[n/2]\log([n/2]) + 7) + n$

$T(n) \leq Cn \log\left(\frac{n}{2}\right) + 14 + n$

$\leq Cn\{\log n - \log 2\} + 14 + n$

$\leq Cn\{\log n - 1\} + 14 + n$

$\leq Cn\{\log n - Cn + 14 + n\}$

$\leq Cn \log n - (C-1)n + 14$

$\leq Cn \log n - b \leq Cn \log n$ if $C \geq 1$

and b is a constant

Hence,

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

Q.29. Consider the problem of adding two n bit binary integers, stored in two n-element array A and B. Sum of the two integers should be stored in binary form in an (n + 1) element array C. State the problem formally and write an algorithm for adding the two integers. **CT : W-11(6M)**

Ans.

```
int main ( )
{
int i;
int A[n];
int B[n];
int C [n + 1];
int f [n] ;
printf ( )\n Enter the n bit no in A");
for (i = 0; i<= 3; i++)
{
    scanf ("%d", A[i]);
}
```

Algorithm for adding the two integer number :

```
Algorithm : ADD Integer (a, b, c)
{
    c : = a + b
    WRITE "C";
}
printf("\n Enter the n bit no in B");
for (i = 0, i<= 3; i++)
{
    scanf ("%d", B[i]);
}
for (i = 0; i<9 ; i++)
    f[i] =0;
for (i=9; i> = 0 ; i--)
{
    C[i] = A [i] + B[i] + f[i];
    if(C[i] > 1)
    {
        f[i - 1] = 1;
        C[i] % = 2;
    }
}
printf ("\n Binary soln is =")
for (i = 0; i < 9; i++)
printf("%i", C[i]);
```

Q.30. Solve the following recurrence relation.

(i) $T(n) = 2T(\sqrt{n}) + \log n$ **CT : S-12, W-13(3M)**

(ii) $T(n) = T(n-1) + \frac{1}{n}$ **CT : S-12(3M)**

Ans.

(i) $T(n) = 2\,T(\sqrt{n}) + \log n$:

Suppose $m = \log_2^n \Rightarrow n = 2^m$

$\therefore n^{1/2} = 2^{m/2} \Rightarrow \sqrt{n} = 2^{m/2}$

put the values, we get

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

Again consider,

$S(m) = T(2^m)$

We have, $S(m) = 2\,S\left(\frac{m}{2}\right) + m$

We known this recurrence has a solution

$S(m) = O(m \log m)$

Substitute the values of m we get,

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

(ii) $T(n) = T(n-1) + \frac{1}{n}$:

By iteration method :

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

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

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

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

$$= \sum_{i=0}^{n-2} \frac{1}{n-i} + T(1) \le \sum_{i=0}^{\infty} \frac{1}{n-i} + \theta(1)$$

Let $n - i = x$

$-di = dx$

Thus, it can be transformed into an integral,

$$\sum_{i=0}^{n} \frac{1}{n-i} = -\int_{n}^{0} \frac{dx}{x} = \log n$$

Thus, $T(n) = 0\,(\log n) + \theta\,(1)$

$\therefore T(n) = \theta\,(\log n)$

SOLUTION OF RECURRENCE RELATION USING TECHNIQUE OF CHARACTERISTIC EQUATION AND GENERATING FUNCTIONS

Q.31. Explain how techniques of characteristic equation can be used to get the solution of recurrence relations. **CT : S-12(7M)**

Ans.

- Recurrence relation can be solved using technique of characteristic equation and generating functions with the resolution of homogenous linear recurrences with constant coefficients i.e. recurrence of the form

$$a_0\, t_n + a_1 t_{n-1} + \ldots\ldots + a_k t_{n-k} = 0$$

This recurrence is :

(1) Linear because it contains terms of form

$t_{n-i}\ t_{n-j},\ t_{n-i}^2$ and so on.

(2) Homogenous because the linear combination of t_{n-i} is equal to zero.

(3) Constant coefficients because the a_i are constants.

Example : $t_n = t_{n-1} + t_{n-2}$

Linear recurrence : A recurrence relation or equation is called a linear recurrence relation or equation if it is a linear combination of order k as given below :

$$a_o\,(n)\,f(n) + a_1\,(n)\,f(n-1)$$
$$+ \ldots\ldots + a_{k-1}\,(n)\,f(n-k+1)$$
$$+ a_k\,(n)\,f\,(n-k) = h\,(n)$$

Homogenous recurrence : If in the recurrence relation there are similar types of terms with same base "b" then such type of recurrence is called as homogenous recurrence.

Example :

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

Non-homogenous recurrence : If there are different terms in the given recurrence which represent different series i.e. if there are multiple type of terms then it is called as non-homogenous recurrence. R.H.S. of non-homogenous recurrence will be non-zero.

Example : $t_n - 2t_{n-1} = 3^n$

Q.32. Solve the given recurrence relations using Master theorem :

(1) $T(n) = 4T\left[\frac{n}{2}\right] + n^2$

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

(3) $T(n) = 4T\left(\frac{n}{2}\right) + n$. **CT : W-12(6M)**

Ans.

(1) $T(n) = 4T\left[\frac{n}{2}\right] + n^2$:

Here, a = 4, b = 2, $f(n) = n^2$

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

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

Now, in this problem $f(n) = n^2$

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

$\therefore \theta(T(n)) = \theta\left(n^{\log_b a} \cdot \log n\right) = \theta(n^2 \log n)$

(2) $T(n) = 4T\left(\frac{n}{2}\right) + n^3$:

Here, a = 4, b= 2, $f(n) = n^3$

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

$\therefore n \log_b a = n^2$

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

$\therefore$ Checking whether $af\left[\frac{n}{b}\right] \leq Cf(n)$.... for any constant C

For many values of C this condition is satisfied

$\therefore \theta(T(n)) = \theta(f(n)) \cdot \theta(n^3)$

(3) $T(n) = 4T\left(\frac{n}{2}\right) + n$:

Here, a = 4, b = 2

$\therefore n^{\log_b a} = n^{\log_2 4} = n^2$

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

$\therefore \theta(T(n)) = \theta(n^{\log_b a}) = (n^2)$

Q.33. $T(n) = T(n/4) + \sqrt{n} + 6$ for $n \geq 4, T(1) = 5$.

Discuss the values for constants.

CT : S-13(8M)

Ans. Step I :

Let $n = 4^i$

$\therefore \sqrt{n} = (4^i)^{\frac{1}{2}} = 2^i$

$\therefore t(i) = t(4^{i-1}) + 2^i + 6$

$\Rightarrow t(i) - t(4^{i-1}) = 2^i + 6$

Step II :

L.H.S $= t(i) - t(4^{i-1})$

Roots (x – 1)

$\therefore r_1 = 1$

R.H.S. $= 2^i + 6$

Roots, $r_2 = 2, r_3 = 1$

Rearrange the roots

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

Roots are real distinct and same.

$t(i) = C_1 . r_1^i + C_2 r_2^i + C_3 . i . r_3^i$

$t(i) = C_1 2^i + C_2 . 1^i + C_3 . i . 1^i$

$t(n) = C_1 \sqrt{n} + C_2 1^{\log_4 n} + C_3 \log_4 n \, 1^{\log_4 n}$ ......(A)

Using given recurrence

$T(n) = T(n/4) + \sqrt{n} + 6$

$T(4) = 13$

$n = 4$ in (A)

$2C_1 + C_2 + C_3 = 13$ ......(B)

At n = 16

$T(16) = 23$

At n = 16 in equation (A)

$4C_1 + C_2 + 2C_3 = 23$ ......(C)

At n = 64

$T(64) = 37$

Put n = 64 in equation (A)

$8C_1 + C_2 + 3C_3 = 37$ ......(D)

Solving equation (B), (C) and (D) we get,

$C_1 = 2, C_2 = 3, C_3 = 6$

Q.34. Solve the following recurrence relation,

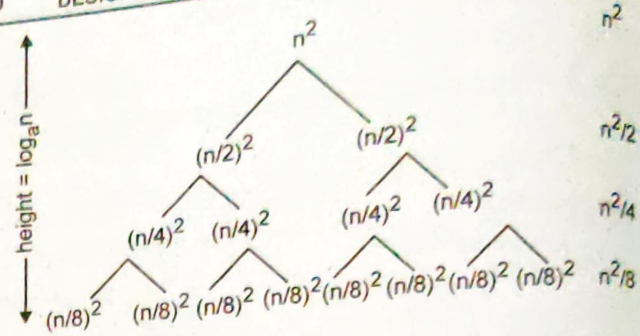
$T(n) = 2T\left(\frac{n}{2}\right) + \theta(n^2)$ using recursion tree method.

CT : S-12(7M)

Ans. $T(n) = 2T\left(\frac{n}{2}\right) + \theta(n^2)$

Using recursion tree method :

- The recursion tree for this recurrence has the following form :



- This is geometric series, thus in the limit the sum is $O(n^2)$. The depth of the tree in this case does not really matter.
- The amount of work at each level is decreasing so quickly that the total is only a constant factor more than root.

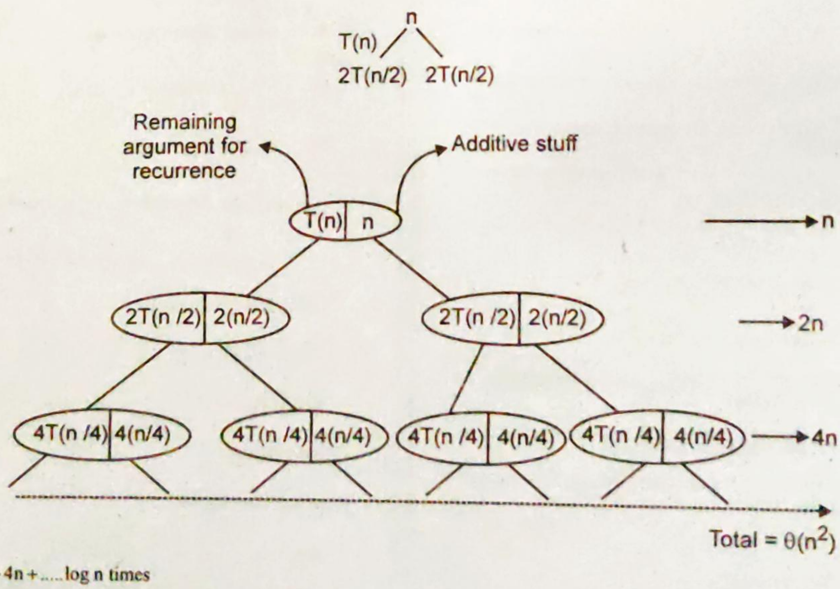
Q.35. Draw the recursion tree for the recurrence.

$T(n) = 4T(n/2) + \theta(n)$ where $C > 0$ and provide a tight asymptotic bound on its solution.

CT : W-13(6M)

Ans. Given : $T(n) = 4T(n/2) + \theta(n)$

We draw recursion tree



$\Rightarrow T(n) = n + 2n + 4n + ...\log n$ times

$T(n) = n(1 + 2 + 4 + ...\log n \text{ times})$

$$T(n) = \frac{n(2^{\log_2 n} - 1)}{2-1}$$

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

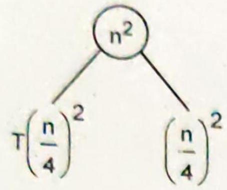
$T(n) = \theta(n^2)$

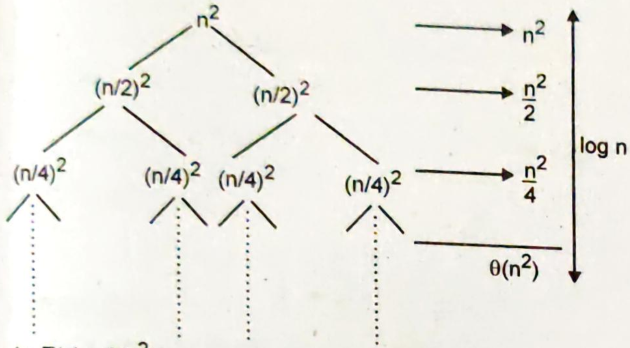
Q.36. Solve the given recurrence using recursion tree method :

$T(n) = 3T(n/4) + \theta(n^2)$.

CS : S-14(7M)

Ans.





i.e $T(n) = \theta(n^2)$

$n + 2n + 4n + \ldots\ldots \log_2 n$ times

$= n\,(1 + 2 + 4 + \ldots\ldots \log_2 n)$ times

$= n\dfrac{(2\log_2 n - 1)}{(2-1)} = \dfrac{n\,(n-1)}{1}$

$= n^2 - n = \theta(n^2)$

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

Q.37. Solve the given recurrence :

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

CT : W-13(3M)

Ans. $t_n - 3t_{n-1} - 4t_{n-2} = 0$

Degree = 2

$x^2 - 3x - 4 = 0$

$\therefore\ x^2 - 4x + x - 4 = 0$

$\therefore\ (x-4)(x+1) = 0$

$\therefore\ r_1 = 4,\ r_2 = -1$

Now,

$t_n = C_1 4^n + C_2 (-1)^n$ ....(2)

Put $n = 0$ in equation (2) we get

$t_0 = C_1 + C_2$ $[\because t_o = 0 \text{ for } n = 0]$

$C_1 + C_2 = 0$ ....(3)

Put $n = 1$ in equation (2)

$4C_1 - C_2 = t_1$

$\therefore\ 4C_1 - C_2 = 5$ $[\because t_1 = 5]$

On solving, we get

$C_1 = 1$ and $C_2 = -1$

$\therefore\ t_n = 4^n - (-1)^n$

Q.38. Solve the following recurrence relation

$t_n = 1\ \forall\ n = 0,\ 2t_{n-1} + n.2^n$ **otherwise**

CS : S-10(7M)

Ans. Step I :

Generate C.E. for recurrence :

$$\begin{array}{ccc} t_n & -2t_{n-1} & = n.2^n \\ \downarrow & \downarrow & \\ 1 & 0 & \end{array}$$

Degree = 1

Step II :

Consider L.H.S,

$$\begin{array}{cc} t_n & -2t_{n-1} \\ \downarrow & \downarrow \\ 1 & 0 \\ (x-2) & = 0 \end{array}$$

$\therefore\ r_1 = 2$

Consider R.H.S,

$n.2^n \Rightarrow b^n.P(n)$

Here, $P(n) = n$

$b^n = 2^n$

i.e. $b = 2$

degree = 1

$\therefore$ Root = twice (d + 1)

$(x-2)(x-2)$

Roots are real and same

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

Step III :

Standard form :

$$t_n = C_1 r_1^n + C_2 . n . r_2^n + C_3 . n^2 . r_3^n$$

$$\therefore\ t_n = C_1 2^n + C_2 . n . 2^n + C_3 . n^2 . 2^n \quad ....(1)$$

Step IV :

Using given recurrence :

$t_0 = 1$ at $n = 0$

Putting $n = 0$, equation (1) will be,

$C_1 = 1$

using given recurrence,

At $n = 1$

$t_1 = 2t_{n-1} + n.2^n$

$\therefore\ t_1 = 2\,t_0 + 1.2^1$

$= 2 + 2$

$= 2 \times 1 + 2$

$\therefore\ t_1 = 4$

At $n = 1$ using equation (1),

$2 + 2\,C_2 + 2\,C_3 = 4$

$\Rightarrow 2C_2 + 2C_3 = 2$

$\Rightarrow C_2 + C_3 = 1 \quad ....(2)$

$C_3 = -C_2$

At $n = 2$

$t_2 = 2t_{n-1} + n.2^n$

$= 2t_{2-1} + 2.2^n$

$= 2t_1 + 8$

$= 2 \times 4 + 8$

$\therefore\ t_2 = 16$

At $n = 2$ using equation (1)

$16 = 4\ C_1 + 8\,C_2 + 16\,C_3$

$\Rightarrow 8\,C_2 + 16\,C_3 = 12 \quad ....(3)$

Also eq^n (2) $\Rightarrow$

$C_2 + C_3 = 1$

$\Rightarrow 8\,C_2 + 8\,C_3 = 8 \quad ....(4)$

On subtracting equation (4) from (3), we get

$$\begin{array}{r} 8\,C_2 + 16\,C_3 = 12 \\ 8\,C_2 + 8\,C_3 = 8 \\ \hline -8\,C_3 = -4 \end{array}$$

$\therefore\ C_3 = \frac{4}{8}$

$\therefore\ C_3 = \frac{1}{2}$

Now,

$C_1 = 1$

$\therefore\ C_2 + C_3 = 1$

$C_3 = -C_2$

$C_2 = \frac{-1}{2}$

$\therefore\ C_1 = 1,\ C_2 = \frac{-1}{2},\ C_3 = \frac{1}{2}$

$\therefore\ t_n = 2^n . 2 - \frac{1}{2} . n . 2^n + \frac{1}{2} . n^2 . 2^n$

Q.39. Solve the given recurrence

$T(n) = mT(n/2) + an^2$ *CT : S-10(7M)*

Ans. **Standard equation :**

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

As value of $a = m$ and $f(n) = an^2$

So, we have to assume value of m.

We consider here value of $m = 7$.

$T(n) = m\,T(n/2) + an^2$

Here, $b = 2$

$f(n) = an^2$

Let assume value of $m = 7$

$b = 2,\ f(n) = an^2$

$n^{\log_b m} = n^{\log_2 7} \approx n^{2.8}$

where $\in = 0.8$

$\therefore\ T(n) = \theta(n \log 7)$

Q.40. Solve the following recurrence relation

(i) $$T(n) = \begin{cases} 1 & n = 1 \\ 2T\left[\frac{n}{2}\right] + n & n > 1 \end{cases}$$

Find an asymptotic bound on T.

(ii) Consider recurrence $T(n) = T(\lfloor n/2 \rfloor + 1)$

Show that it is asymptotically bound by O (log n).

CT : S-10(8M)

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

CT : W-13(3M)

Ans.

(i) **Given :** $T(n) = \begin{cases} 1 & n = 1 \\ 2T\left[\frac{n}{2}\right] + n & n > 1 \end{cases}$

- We guess the solution in O (n log n). Thus for a constant 'C'.

$T(n) \le Cn \log n$

Put this in the given recurrence equation :

Now,

$$T(n) \le 2C\left[\frac{n}{2}\right]\log\left[\frac{n}{2}\right] + n$$

$\le Cn \log n - Cn \log 2 + n]$

$= Cn \log n - n(C \log 2 - 1)$

$\le Cn \log n, \forall C \ge 1$

- By mathematical induction, we require to show that our solution holds for the boundary conditions.

$T(1) \le C.1 \log 1 = 0$

Thus, for any value of C, this will not hold, so by asymptotic definition we need to prove

$T(n) \le Cn \log n$ for all $n \ge n_0$

Now, $T(2) \le C.2 \log 2$

$T(3) \le C.3 \log 3$

Thus, the above relation holds for $C \ge 2$.

- Hence our guess $T(n) = O(n \log n)$ is true.

For most recurrences we shall examine, it is straight forward to extend boundary conditions to make inductive assumption work for small n.

(ii) $T(n) = T\left(\left\lfloor\frac{n}{2}\right\rfloor\right) + 1$

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

We have to show that for some constant C,

$T(n) \le C \log n$

Put this in the given recurrence equation

= quadratic function of n

$= O(n^2)$

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

Put $m = \log n$

$n = 2^m$

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

Renaming

$S(m) = S\left(\frac{m}{2}\right) + 1$ .....(2)

$[\because S(m) = T(2^m)]$

Compare equation (2) with

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

we get,

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

$m^{\log_b a} = m^{\log_2 1} = m^0 = 1$

$= \theta(1)$

case 2 of master theorem is satisfied.

$S(m) = \theta(m^{\log_b a} \log m)$

$S(m) = \theta(1. \log m)$

$S(m) = \theta(\log \log n)$ $[\because m = \log^n]$

$T(n) = T(2^m) = S(m) = \theta(\log \log n)$

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

Q.41. Using master method solve

$$T(n) = 3T\left(\frac{n}{4}\right) + \sqrt{n} + 5 \forall n \ge 4$$

$T(i) = 2$

CS : S-10(7M), W-12(9M)

Ans. Step 1 :

Let $n = 4^i$; $i = \log_4 n$

$\sqrt{n} = (4^i)^{1/2} = 2^i$

$t(i) = 3t(4^{i-1}) + 2^i + 5$

$\therefore t(i) - 3t(4^{i-1}) = 2^i + 5$

Step 2 :

L.H.S = (x − 3)

$\therefore r_1 = 3$

R. H. S = $2^i + 5 \Rightarrow (x-2)$ and $(x-1)$

$\therefore r_1 = 3, \ r_2 = 2, \ r_3 = 1$

Step 3 :

Roots are real and distinct.

$t(i) = C_1 r_1^i + C_2 r_2^i + C_3 r_3^i$

$n = 4^i$; $i = \log_4 n$

$t(i) = C_1 3^i + C_2 2^i + C_3 1^i$

$t(i) = C_1 3^i + C_2 \sqrt{n} + C_3$ ......(A)

using given recurrence

$T(n) = 3T(n/4) + \sqrt{n} + 5$

$T(4) = 3T(1) + 2 + 5$ $[\because T(1) = 2]$

$= 3 \times 2 + 2 + 5$

$= 6 + 2 + 5$

$\therefore T(4) = 13$

Put n = 4 in (A)

$\therefore t(4) = C_1 3^{\log_4 4} + C_2 \sqrt{4} + C_3$

$\therefore 13 = C_1 3 + 2C_2 + C_3$ $[\because \log_4^4 = 1]$

$\therefore 3C_1 + 2C_2 + C_3 = 13$ ......(B)

At n = 16

$T(16) = 3T(16/4) + \sqrt{16} + 5$

$= 3T(4) + 4 + 5$

$= 3 \times 13 + 4 + 5$

$= 39 + 4 + 5$

$\therefore T(16) = 48$

Put n = 16 in (A)

$\therefore T(16) = C_1 3^{\log_4 16} + C_2 \sqrt{16} + C_3$

$\therefore 48 = 9C_1 + 4C_2 + C_3$

$\therefore 9C_1 + 4C_2 + C_3 = 48$ ......(C)

At n = 64

$T(64) = 3T(64/4) + \sqrt{64} + 5$

$= 3 \times 16 + 8 + 5$

$= 48 + 8 + 5$

$= 61$

Put n = 64 in (A)

$\therefore T(64) = C_1 3^{\log_4 64} + C_2 \sqrt{64} + C_3$

$= C_1 . 3^3 + 8C_2 + C_3$

$\therefore 27C_1 + 8C_2 + C_3 = 61$ .....(D)

Solving equation B, C and D, we get

$C_1 = -9.5$

$C_2 = 46$

$C_3 = -50.5$

Putting values in equation (A), we get

$t(n) = -9.5 \times 3^{\log_4 n} + 46\sqrt{n} - 50.5$

Q.42. Using master method solve the following recurrence and also find the values of constant involved.

$$T(n) = T\left(\frac{n}{4}\right) + \sqrt{n} + 4 \text{ for } n \geq 4 \text{ and } T(1) = 4.$$

CS : W-10(7M), W-11(5M), W-13(6M)

Ans. **Step I :**

Let $n = 4^i$

$\therefore \sqrt{n} = (4^i)^{1/2} = 2^i$

$t(i) = t(4^{i-1}) + 2^i + 4$

$\therefore t(i) - t(4^{i-1}) = 2^i + 4$

Step II :

L.H.S = T (n) – T (n/4)

$\therefore (x - 1)$

$\therefore r_1 = 1$

R. H. S = $\sqrt{n} + 4$ i.e. $2^i + 4$

$\therefore (x-2)(x-1)$

$\therefore r_2 = 1, \ r_3 = 1$

Step III :

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

Roots are real, distinct and same

$\therefore t(i) = C_1 r_1^i + C_2 r_2^i + C_3 r_3^i$

$n = 4^i$; $i = \log_4^n$

$\therefore t(i) = C_1 2^i + C_2 1^i + C_3 i.1^i$

$\therefore t(i) = C_1 2^i + C_2 + C_3 i$

$\therefore t(n) = C_1 \sqrt{n} + C_2 + C_3 . \log_4^n$ ......(A)

using given recurrence

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

$T(4) = T(1) + 2 + 4$

$= 4 + 2 + 4$

$\therefore T(4) = 10$

Put n = 4 in (A)

$\therefore\ 2C_1 + C_2 + C_3 = 10$ ......(B)

At n = 16

$T(16) = T(16/4) + \sqrt{16} + 4$

$= T(4) + 4 + 4$

$= 10 + 4 + 4$

$\therefore\ T(16) = 18$

Put n = 16 in (A)

$\therefore\ T(16) = 4C_1 + C_2 + 2C_3$

$\therefore\ 4C_1 + C_2 + 2C_3 = 18$ ......(C)

At n = 64

$T(64) = T(64/4) + 8 + 4$

$= T(16) + 8 + 4$

$= 18 + 8 + 4$

$T(64) = 30$

Put n = 64 in (A)

$\therefore\ 8C_1 + C_2 + 3C_3 = 30$ .....(D)

Solving equation (B), (C) and (D), we get

$C_1 = 2,\ C_2 = 2,\ C_3 = 4$

$\therefore\ t_n = 2\sqrt{n} + 2 + 4\log_4^n$

Q.43. Solve the following :

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

CS : W-10(4M)

Ans. **Step I :**

Generate recurrence

$t_n - 3t_{n-1} = 2^n + n$

Step II :

Consider L. H. S,

$t_n - 3t_{n-1}$

$(x - 3) = 0$

$\therefore\ r_1 = 3$

Consider R. H. S,

$2^n + n \Rightarrow b_1^n . P_1(n) + b_2^n . P_2(n)$

$b_1^n . P_1(n) = 2^n$

$\Rightarrow b_1^n = 2^n$ or $b = 2$

$P_1(n) = 1$ degree, $d = 0$

Root = once i.e d + 1

$r_2 = 2$,

$b_2^n . P_2(n) = n$

$\therefore\ b_2^n = 1$ or $b = 1$

$P_2(n) = n$ degree, $d = 1$

$\therefore$ Root will be repeated for d + 1 times i.e. root will be twice

$(x - 1)\ (x - 1)$

$r_3 = 1,\ r_4 = 1$

$\therefore\ r_1 = 3,\ r_2 = 2,\ r_3 = 1,\ r_4 = 1$

Step III :

Standard solution :

Roots are real, distinct and same

$\therefore\ t_n = C_1 r_1^n + C_2 r_2^n + C_3 r_3^n + C_4 . n . r_4^n$ .....(i)

Step IV :

Using given recurrence

$t_0 = 2$

At n = 0 equation (1) will be

$\therefore\ 3C_1 + 2C_2 + C_3 = 2$ ......(A)

Using given recurrence

At n = 1

$t_n = 3t_{n-1} + n + 2^n$

$\therefore\ t_1 = 3t_0 + 1 + 2$

$= 3 \times 2 + 1 + 2$ $[\because t_o = 2]$

$\therefore\ t_1 = 9$

At n = 1 in equation (1), we have,

$t_1 = 3C_1 + 2C_2 + C_3 + C_4$

$\therefore\ 3C_1 + 2C_2 + C_3 + C_4 = 9$ .....(B)

At n = 2 in given recurrence

$t_2 = 3t_1 + 2 + 4$

$= 3 \times 9 + 2 + 4$

$= 27 + 6$

$= 33$

At n = 2 in equation (1), we get

$t_2 = 9C_1 + 4C_2 + C_3 + 2C_4$

$\therefore\ 9C_1 + 4C_2 + C_3 + 2C_4 = 33$ .....(C)

At n = 3 in given recurrence

$\therefore\ t_3 = 3t_2 + 3 + 8$

$= 3 \times 33 + 3 + 8$

$= 99 + 3 + 8$

$\therefore\ t_3 = 110$

At $n = 3$ in equation (1), we have,

$t_3 = 27\,C_1 + 8\,C_2 + C_3 + 3\,C_4$

$\therefore\ 27\,C_1 + 8\,C_2 + C_3 + 3\,C_4 = 110$ .....(D)

Solving equation (a), (b), (c) and (d), we get

$C_1 = 5.6,\ C_2 = -8,\ C_3 = 0,\ C_4 = 7$

Q.44. Explain logarithmic recurrences. Solve the problem by using it :

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

CT : S-14(6M)

OR Define logarithmic recurrence with suitable example.

CS : W-10(3M)

Ans. Logarithmic recurrence :

- In a algorithm a index variable is operated by either multiplication or division operator then the algorithm will generate recurrence called as logarithmic recurrence.
- To solve the recurrence it is necessary to remove the division operator. This is possible by changing the base of recurrence.

Standard form :

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

Where, a = No. of recursive calls

b = Division factor for input

f (n) = Any function

Example :

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

Step I :

Let $n = 2^i$; $i = \log_2 n$

Then recurrence can be written as :

$t_n = 3\,t(2^i/2) + 2^i$

$\therefore\ t_n = 3\,t(2^{i-1}) + 2^i$

$\therefore\ t_n - 3\,t(2^{i-1}) = 2^i$ .....(1)

Step II :

Equation (1) is non-homogenous finding roots of L.H.S and R.H.S

$L.H.S = t_n - 3\,t(2^{i-1})$

Root is $(x - 3)$

$\therefore\ r_1 = 3$

$R.H.S = 2^i \Rightarrow b^i\,P(i)$

$b = 2,\ P(i) = 1,\ d = 0$, Root = once

$(x - 2)$ i.e. $r_2 = 2$

$b_1^n \cdot P_1(n) = 2^n$

Step III :

Standard solution :

Roots are real and distinct

$\therefore\ t(i) = C_1\,r_1^i + C_2\,r_2^i$

$\therefore\ t(i) = C_1\,3^i + C_2\,2^i$

Step IV :

Generating solution for n^{th} term

$n = 2^i$

$i = \log_2 n$

$t(n) = C_1 \cdot 3^{\log_2 n} + C_2 \cdot n$ .....(2)

$T(1) = 1$, if $n = 1$

Using equation (2)

$T(1) = C_1 \cdot 3^{\log_2 1} + C_2$

$= C_1 \cdot 3^0 + C_2$

$\therefore\ 1 = C_1 + C_2$

$\therefore\ C_1 + C_2 = 1$ .....(3)

Using given recurrence for n = 2

$\therefore\ T(2) = 3\,T(2/2) + 2$

$= 3\,T(1) + 2$

$\therefore\ T(2) = 5$

Using equation (2)

$T(2) = 3C_1 + 2C_2$

$\therefore\ 3C_1 + 2C_2 = 5$ .....(4)

Solving equation (3) and (4), we get

$C_1 = 3, \ C_2 = -2$

$\therefore$ Complete solution is :

$t(n) = 3 \,.\, 3^{\log_2 n} - 2n$

Q.45. Solve the following recurrence :

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

for $n \geq 4$ $T(1) = 3$

CS : S-11(8M)

Ans. **Step I :**

Let $n = 4^i$

Then recurrence can be written as :

$\therefore \sqrt{n} = (4^i)^{1/2} = 2^i$

$\therefore t(i) = 2t(4^{i-1}) + 2^i + 3$

$\therefore t(i) - 2t(4^{i-1}) = 2^i + 3$

Step II :

$L.H.S = t(i) - 2t(4^{i-1})$

Root is $(x-2)$ i.e. $r_1 = 2$

$R.H.S = 2^i + 3$

Roots are $(x-2)\ (x-1)$

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

Rearrange the roots, same roots should be last

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

Step III :

Roots are real and distinct and same standard form :

$\therefore t(i) = C_1 r_1^i + C_2 r_2^i + C_3 . i . r_3^i$

$n = 4^i$

$\therefore i = \log_4 n$

$\therefore t(i) = C_1 1^i + C_2 2^i + C_3 . \log_4^n . 2^i$

$\therefore t(i) = C_1 1^i + C_2 \sqrt{n} + C_3 \sqrt{n} . \log_4 n$ ...(A)

Using given recurrence,

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

At $n = 4$

$T(4) = 2T(4/4) + \sqrt{4} + 3$

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

$= 2 \times 3 + 2 + 3$ $[\because T(1) = 3]$

$\therefore T(4) = 11$

Put $n = 4$ in equation (A)

$T(4) = C_1 + 2C_2 + 2C_3$

$\therefore C_1 + 2C_2 + 2C_3 = 11$ ....(B)

At $n = 16$

$T(16) = 2T(16/4) + \sqrt{16} + 3$

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

$= 2 \times 11 + 4 + 3$

$\therefore T(16) = 29$

Put $n = 16$ in equation (A)

$\therefore T(16) = C_1 + 4C_2 + 8C_3$

$\therefore C_1 + 4C_2 + 8C_3 = 29$ ....(C)

At $n = 64$

$T(64) = 2T(64/4) + 8 + 3$

$\therefore T(64) = 43$

Put $n = 64$ in equation (A)

$\therefore C_1 + 8C_2 + 24C_3 = 43$ ....(D)

Solving equation (B), (C) and (D), we get

$C_1 = -29, \ C_2 = 25.5, \ C_3 = -5.5$

Q.46. $T(n) = 3$ if $n = 0$

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

CS : S-11(6M)

Ans. **Step I :**

Generate recurrence :

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

Step II :

Consider L. H. S,

$t_n - 2t_{n-1}$

Root is $(x-2)$

$\therefore r_1 = 2$

Consider R. H. S,

$2^n + 5 \Rightarrow b_1^n . P(n) + b_2^n . P(n)$

$b_1^n . P_1(n) = 2^n$; $b = 2$; $P_1(n) = 1$, $d = 0$

$\therefore$ Root once $(d+1)$

$(x-2)$ i.e. $r_2 = 2$

$b_2^n . P_2(n) = 5$

Here, $b_2 = 1$, $P_2(n) = 5$, $d = 1$

$\therefore$ Root twice (d + 1)

$(x-1)(x-1)$

$r_3 = 1$, $r_4 = 1$

Step III :

Standard solution :

Roots are real, distinct and same.

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

$\therefore t_n = C_1 r_1^n + C_2 r_2^n . n + C_3 r_3^n + C_4 . n . r_4^n$

$\therefore t_n = C_1 . 2^n + C_2 . n . 2^n + C_3 . 1^n + C_4 . n . 1^n$ .....(1)

Step IV :

Using given recurrence

T (n) = 3

At n = 0 equation (1), we have

$\therefore C_1 + C_3 = 3$ ......(A)

Using given recurrence

At n = 1

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

$\therefore t_1 = 2t_o + 2 + 5$

$= 2 \times 3 + 2 + 5$

$\therefore t_1 = 13$

At n = 1 in equation (1)

$\therefore t_1 = 2C_1 + 2C_2 + C_3 + C_4$

$\therefore 2C_1 + 2C_2 + C_3 + C_4 = 13$ .....(B)

At n = 2

$t_2 = 2t_1 + 2^n + 5$

$= 2 \times 13 + 4 + 5$

$= 26 + 4 + 5$

$\therefore t_2 = 35$

At n = 2 in equation (1)

$t_2 = 4C_1 + 8C_2 + C_3 + 2C_4$

$\therefore 4C_1 + 8C_2 + C_3 + 2C_4 = 35$ .....(C)

At n = 3

$\therefore t_3 = 2t_2 + 8 + 5$

$= 2 \times 35 + 8 + 5$

$\therefore t_3 = 83$

At n = 3 in equation (1)

$\therefore t_3 = 8C_1 + 24C_2 + C_3 + 3C_4$ .....(D)

$8C_1 + 24C_2 + C_3 + 3C_4 = 83$

Solving equation (A), (B), (C) and (D), we get the values C_1, C_2, C_3 and C_4

$C_1 = 8$, $C_2 = 1$, $C_3 = -5$, $C_4 = 0$

Q.47. Solve the following recurrence : CS : W-11(6M)

$$t_n = \begin{cases} 1 & \text{if } n = 0 \\ 2T(n/2) + n & \text{otherwise} \end{cases}$$

Ans. Step I :

Let $n = 2^i$; $i = \log_2 n$

Then recurrence can be written as :

$$t_n = 2t\left(\frac{2^i}{2}\right) + 2^i$$

$\therefore t_n = 2t(2^{i-1}) + 2^i$

$\therefore t_n - 2t(2^{i-1}) = 2^i$ .....(1)

Step II :

Equation (1) is non-homogenous finding roots of L.H.S and R.H.S

L. H. S = $t_n - 2t(2^{i-1})$

$\therefore (x-2)$

$\therefore r_1 = 2$

R. H. S = $2^i \Rightarrow b^i P(i)$

$b = 2$, $P(i) = 1$, $d = 0$

$\therefore (x-2)$

$\therefore r_2 = 2$

$\therefore r_1 = 2$, $r_2 = 2$

Roots are real and same.

Step III :

$t(n) = C_1 r_1^n + C_2 . n . r_2^n$

$\therefore t(i) = C_1 r_1^i + C_2 . i . r_2^i$

$= C_1 2^i + C_2 . i . 2^i$

$\therefore t(n) = C_1 n + C_2 . \log_2 n . n$ ....(2)

Using given recurrence and n = 0, we have

T (n) = 2 T (n/2) + n

$\therefore$ T (0) = 2 T (0/2) + 0

= 2 T (0) + 0

$\therefore$ T (0) = 2

But T(0) = 1 is given

Put n = 1 in equation 2

$T(1) = C_1$

The above problem is not solvable because value of $t_n = 1$ if n = 0 given ; if n = 1 is given then the problem is solvable.

Q.48. Use Master method to give tight asymptotic bound for the following recurrences :

(i) T (n) = 3 T (n/4) + n log n

(ii) T (n) = 16 T (n/4) + n^2.

CS : S-12(7M)

Ans.

(i) **T (n) = 3 T (n/4) + n log n :**

Here, a = 3, b = 4, f(n) = n log n

$\therefore n^{\log_b a} = n^{\log_4 3} = 0(n^{0.793})$

$f(n) = \Omega(n^{\log_4^{3+\epsilon}})$

where $\epsilon = 0.2$

Case 3 applies, now for regularity condition i.e.

$$a f\left(\frac{n}{b}\right) \le c f(n)$$

$$\Rightarrow 3\left(\frac{n}{4}\right)\log\left(\frac{n}{4}\right) \le \left(\frac{3}{4}\right) n \log n$$

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

$\therefore T(n) = \theta(n \log n)$

(ii) **T (n) = 16 T (n/4) + n^2 :**

Here, a = 16, b = 4 and f(n) = n^2

Now, we find $n^{\log_b^a}$ to apply cases for master theorem

$$n^{\log_b a} = n^{\log_4 16} = n^{\log_4 4} = n^{2\log_4 4}$$

$$= n^2 . 1$$

$$= n^2$$

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

$$= \theta(n^2)$$

This shows that case 2 of master theorem is satisfied.

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

$\therefore T(n) = \theta(n^2 \log n)$

Q.49. Solve the following recurrence :

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

CS : W-12(4M), W-13(8M)

OR Design the recurrence relation and solve using master method for the following :

There are two recursive calls in a "for loop" to same function in which input is divided by factor of 2. There is one more call to main function which operates on the "n" integers. The process terminates at n = 1.

CS : S-13(7M)

Ans. Let $n = 2^i$

$\therefore i = \log_2^n$

$t_n = 2t(2^i/2) + 2^i$

$\therefore t_n - 2t(2^{i-1}) = 2^i$

L. H. S = $t_n - 2t(2^{i-1})$

$\therefore (x-2)$

$\therefore$ Root $r_1 = 2$

R. H. S = 2^i

$\therefore (x-b)$; Here b = 2

$\therefore r_2 = 2$

Roots are real and same.

$$t(i) = C_1 r_1^i + C_2 . i . r_2^i$$

$$= C_1 2^i + C_2 . \log_2^n . 2^i$$

$$\therefore t(n) = C_1 n + C_2 . n . \log_2^n \quad \text{..... (1)}$$

Given, T (1) = 1

Using equation (1), we have

$T(1) = C_1$ i.e. $C_1 = 1$

For n = 2

T (2) = 2 T (1) + 2 = 4

$\therefore T(2) = 2 C_1 + 2C_2 = 4$

$2C_2 = 2$

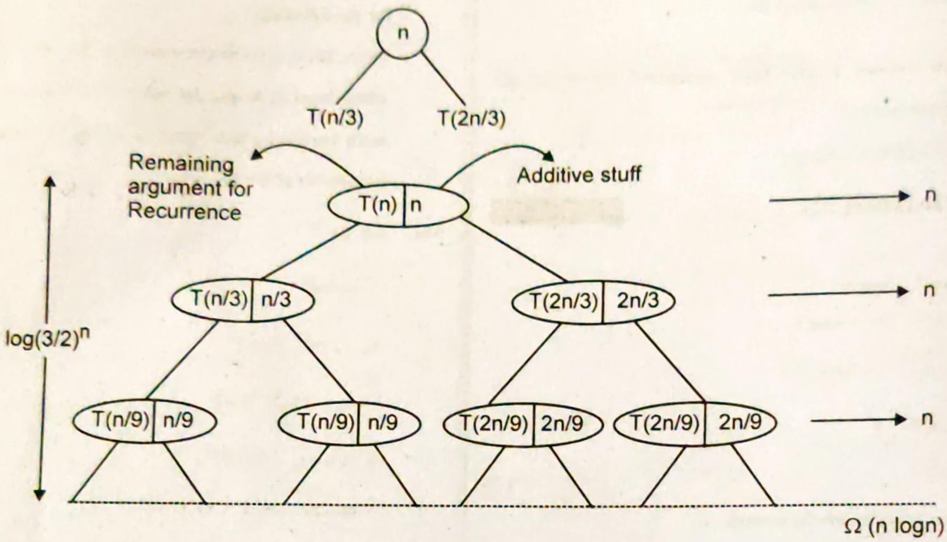
$\therefore C_2 = 1$

$\therefore C_1 = 1$ and $C_2 = 1$

Q.50. Draw the recursion tree for T (n) = T (n/3) + T (2n/3) + O (n) and find out the upper bound for the recursion. Also comment on value generated using upper bound. **CS : S-13(6M)**

Ans. $T(n) = T(n/3) + T\left(\frac{2n}{3}\right) + O(n)$

Now, we draw the recursion tree



$T(n) = n + n + n + ... + \log_{\frac{3}{2}} n$ times

$\therefore\ T(n) = \Omega(n \log n)$

COMPLEXITY CALCULATIONS OF VARIOUS STANDARD FUNCTIONS

Q.51. Define time and space complexity. **CT : W-10(3M)**

Ans.

- It is very convenient to classify algorithm based on the relative amount of time or relative amount of space required and specify the growth of time/space requirements as a function of the input size.

(1) Time complexity :

Running time of the program as a function of the size of input is called as time complexity.

(2) Space complexity :

- Space complexity means the memory required for the program to run to completion.
- The space complexity analysis was critical in the early day's of computing when storage space on a computer (both internal and external) was limited.
- When considering space complexity algorithms are divided into those that need extra space to do their work and those that work in place.
- It was that not unusual for programmers to choose an algorithm that was slower just because it worked in place, because there was not enough extra memory for a faster algorithm.

Q.52. How would we calculate running time of an algorithm? **CT : W-10(6M)**

Ans.

- We can calculate the running time of an algorithm reliably by running the implementation of the algorithm on a computer. Alternatively we can calculate the running time by using a technique called algorithm analysis.

- We can estimate an algorithm's performance by counting the number of basic operations required by the algorithm to process an input a certain size.

Basic operation :

- The time to complete a basic operation does not depend on the particular values of its operands. So it takes a constant amount of time.

Example : Arithmetic operation, Boolean operation, Comparison operation, Module operation, Branch operation.

Input size : It is the number of input processed by the algorithm.

- **Example** : For sorting algorithm the input size is measured by number of records to be sorted.

Q.53. Consider following for loops. Calculate the total computation time for the following :

```
for i ← 2 to m – 1
for j ← 3 to i
    {
        sum = sum + A [i] [j]
    }
```

CT : W-10(3M)

Ans. The total computational time is :

$$\sum_{i=2}^{m-1}\sum_{j=3}^{i} t_{ij} = \sum_{i=2}^{m-1}\sum_{j=3}^{i} \theta(1)$$

$$= \sum_{i=2}^{m-1} \theta(1)$$

$$= \theta\left(\sum_{i=2}^{m-1} i\right)$$

$$= \theta(m^2/2 + \theta(m))$$

$$= \theta(m^2).$$

Q.54. Find the time complexity for the following :

```
(i) I = 0
    for j = 1 to n do
    {
        for i = 1 to n² do
        {
            for K = 1 to n³ do
            {
                I = I + 1
            }
        }
    }

(ii) I = 0
    for i = 1 to n do
    {
        for j = i + 2 to n do
        {
            for K = 1 to j – 1 do
            {
                I = I + 1
            }
        }
    }
```

CT : W-06(5M)

Ans.

(i) Time complexity is $O(n^3)$.

(ii) Time complexity is O (n).

Q.55. Define growth rate of function. CT : W-10(3M)

Ans. Growth rate :

- The growth rate of an algorithm is the rate at which the running time (cost) of the algorithm grows as the size of input grows.
- The growth rate has a tremendous effect on the resources consumed by the algorithm.
- Resources for an algorithm are usually expressed as a function of input. Often this function is messy and difficult to work.
- To study function growth easily, we reduce the function down to the important part.

  Let $f(n) = an^2 + bn + c$

- In this function, the n^2 term dominates the function that is when n gets sufficiently large, the other terms bare factor into the result.
- Dominate terms are what we are interested in to reduce a function, in this we ignore all constants and coefficients and look at the highest order term in relation to n.

PRINCIPLES OF DESIGNING ALGORITHM

Q.56. Define algorithm in detail. Explain their four distinct area of study. CT : S-14(6M)

Ans. Algorithm :

- An algorithm is a set of rules for carrying out calculation either by hand or on a machine.
- An algorithm is a sequence of computational steps that transform the input into the output.
- An algorithm is an abstraction of program to be executed on a physical machine.

The four distinct area of study are as follows :

(1) **To devise an algorithm :** This is an activity where human intelligence is definitely required ; some of the strategies used have a general applicability like dynamic programming, divide and conquer.

(2) **To express an algorithm :** An algorithm can be expressed in various ways such as flowchart, pseudo-code, program.

(3) **To validate an algorithm :** This means that a program satisfies its precise specifications.

(4) **To analyze an algorithm :** This field of study is called analysis of algorithm. It is concerned with the amount, computer time and storage that is required by the algorithm.

(5) **To test a program :** Testing a program consists of two parts - debugging and profiling. Profiling is a process of executing a correctly working program with the appropriate data sets.

Q.57. Define different design approaches of an algorithm. CT : W-10(7M)

Ans. Approaches to design an algorithm :

There are basically two approaches for designing an algorithm as follows :

(1) Incremental approach.

(2) Divide and conquer approach.

(1) Incremental approach :

- In this approach every time we increase the index to insert the element in proper place.
- "Job is partly done - do a little more, repeat, until done."
- A good example of this approach is "Insertion sort."

(2) Divide and conquer approach :

- Divide and conquer is a recursive approach. This approach works as follows :

(i) Divide problem into sub-problems of the same kind of problem.

(ii) For sub-problems that are really small (trivial) solve them directly. Else solve them recursively (conquer).

(iii) Combine sub-problem solution to solve the whole things.

- A good example of this approach is "Merge Sort."

Q.58. Suppose the following algorithm is used for evaluating the polynomial :

$p(x) = a_n x_n + a_{n-1} x_{n-1} + a_1 x + a_0$

p : a_0 ;

pow x = 1 ;

for i : = 1 to n do

pow x : = x * pow x

p : p + a_1 * pow x

end.

(i) How many multiplications and additions are done in worst case?

(ii) How many multiplications are done in average case?

(iii) How can you improve this algorithm?

Ans. Given :

$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_1 x + a_0$

P : a_0

POW x = 1 ;

For i : = 1 to n do

POW x : = x * POW x

P : P + a_i * POW x

End

$\Rightarrow P(x) = a_n x^n + a_{n-1} x^{n-1} + a_1 x + a_0$

is the univariate polynomial equation.

Where, x is an indeterminate and a_i may be integers floating point numbers or more generally elements of a commutative ring or a field. If $a_n \neq 0$ then n is called the degree of A.

(i) The total number of multiplication done in worst case is 2n and addition is n.

(ii) The total number of multiplication done in average case is also 2n.

(iii) To improve this algorithm we can use W.G. Horner improvement technique in 1819 to evaluate the coefficient of A (x + C)

Algorithm Horner (A, n, v)

```
{
    P : a_n ;
    for i : = n – 1 to 0 step 1 – 1 do
    {
        P : P * v + a_i ;
    }
    return P;
}
```

POINTS TO REMEMBER :

(1) *Induction is a method to prove sequence of steps. The two phases of induction are :*

(i) Basic phase

(ii) Hypothesis phase.

(2) *An arithmetic series is the sum of a sequence $\{a_k\}$, k = 1, 2, ... in which each term is computed from the previous one by adding or subtracting a constant d. Therefore, for k > 1*

$$a_k = a_{k-1} + d = a_{k-2} + 2d + ...a_1 + d(k + 1)$$

(3) *In geometric series the ratio of successive terms are constant, so that the series may be represented as : $a, ar, ar^2, ...ar^{n-1}$.*

(4) *A recurrence relation is an equation which is defined in terms of itself.*

(5) *The five methods for solving recurrence are :*

(i) Substitution method.

(ii) Iteration method.

(iii) Recursion tree.

(iv) Changing variables.

(v) Master theorem.

(6) *Running time of the program as a function of the size of input is called as time complexity.*

(7) *The memory required for the program to run to completion is called as space complexity.*

(8) *The growth rate of an algorithm is the rate at which the running time (cost) of the algorithm grows as the size of input grows.*

(9) *An algorithm is a sequence of computational steps that transforms the input into the output.*

(10) *Two approaches for designing an algorithm are :*

(i) Incremental approach.

(ii) Divide and conquer approach.