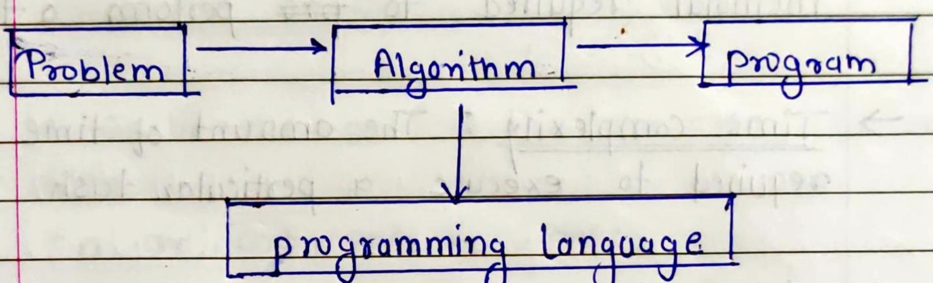
Algorithm :- An algorithm is any well-defined computational procedure that takes some value or set of value as input and produces some value of set of values as output.

An algorithm is a sequence of computational steps that program or transform input into output.



The study of algorithm is called algorithmic

★ Properties of algorithm

i) **Input output :** Each algorithms is supplied with zero or more external quantities and also produces at least one quantities as a output.

ii) **Definitiness :** Each instruction of an algorithm must be clear and unambitious

iii) Finitness : Algorithm ~~tt~~ must terminate after finite number of steps.

iv) performance : The performance of algorithm is measured in terms of space and time complexity.

→ Space complexity : The total amount of memoral required to ~~pro~~ perform a task.

→ Time complexity : The amount of time required to execute a perticular task.

★ Arithmetic series :

Let S_n be the sum of first n terms of arithmatic series

$a ; a+d , a+2d , a+3d , \ldots , a+(n-1)d$

then,

$$S_n = an + \frac{n(n-1)d}{2}$$

$$S_n = a + a+d + \ldots + a+(n-2)d + a+(n-1)d \quad — ①$$

$$S_n = a+(n-1)d + a+(n-2)d + \ldots + a+d + a \quad — ②$$

Adding ① & ②

$$2S_n = a + a+(n-1)d + a+d + a+(n-2)d + \ldots + a+(n-2)d + a+d + a+(n-1)d + a$$

$$2S_n = 2a + (n-1)d + 2a + (n-1)d + \dots + 2a + (n-1)d + 2a + (n-1)d.$$

$$2S_n = n[2a + (n-1)d]$$

$$S_n = \frac{\cancel{2}an}{\cancel{2}} + \frac{n(n-1)}{2}d$$

$$S_n = an + \frac{n(n-1)d}{2}$$ Hence proved

- Geometric series

Let S_n be the sum of first n term of geometric series

$a, ar, ar^2, ar^3, \dots, ar^{n-1}$

then,

$$\boxed{S_n = \frac{a(1-r^n)}{(1-r)}}$$

$$S_n = a + ar + ar^2 + \dots + ar^{n-1} \quad \text{— (1)}$$

Multiply eq (1) by r

$$rS_n = ar + ar^2 + ar^3 + \dots + ar^n \quad \text{— (2)}$$

~~Sn~~ Subtracting eq (1) & (2)

$$S_n - rS_n = (a + ar + ar^2 + \dots + ar^{n-1}) - (ar + ar^2 + ar^3 + \dots + ar^n)$$

$$(1-r)S_n = a - ar^n$$

$$\boxed{S_n = \frac{a(1-r^n)}{(1-r)}}$$ Hence proved.

$(k-1)a^{(k-1)} - (k-2)a^{(k-1)}$

$ka^{\cancel{k-1}} - a^{k-1} - ka^{k-1} + 2a^{k-1}$

$\boxed{a^{k-1}}$

1. Derive closed form solution for given

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

$\Rightarrow$ $S_k = a + 2a^2 + 3a^3 + \dots + (k-1)a^{(k-1)}$ — ①

multiply eq ① by a

$aS_k = a^2 + 2a^3 + 3a^4 + (k-1)a^k$ — ②

① - ②

$S_k - aS_k = (a + 2a^2 + 3a^3 + \dots + (k-1)a^{(k-1)}) - (a^2 + 2a^3 + 3a^4 + \dots (k-1)a^k)$

$(1-a)S_k = a + a^2 + a^3 + a^4 + \dots + a^{k-1} - (k-1)a^k$

$= a + a^2 + a^3 + a^4 + \dots + a^{k-1} - ka^k + a^k$

$= a + a^2 + a^3 + a^4 + \dots + a^{k-1} + a^k - ka^k$

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

$(1-a)S_k = a\left[\frac{1(1-a^k)}{(1-a)}\right] - ka^k$

$(1-a)S_k = \frac{a(1-a^{(k)})}{(1-a)} - k.a^k$

divide by $(1-a)$

$S_k = \frac{a(1-a^k)}{(1-a)^2} - \frac{k.a^k}{(1-a)}$

$k2^k - (k-1)2^k$
$k2^k - \cancel{k2^k} + 2^k$
2^k

Page No.
Date

2. Find closed form solution for given summation.

i.e.

$$\sum_{i=1}^{k} i.2^i$$

$\Rightarrow$ $S_k = 2 + 2.2^2 + 3.2^3 + \dots + k.2^k$ — (1)

Multiply eq (1) by 2

$2S_k = 2.2 + 2.2^3 + 3.2^4 + \dots + k.2^{k+1}$

$2S_k = 2^2 + 2.2^3 + 3.2^4 + \dots + k.2^{k+1}$ — (2)

(1) - (2)

$S_k - 2S_k = (2 + 2.2^2 + 3.2^3 + \dots + k.2^k) - (2^2 + 2.2^3 + 3.2^4 + \dots + k.2^{(k+1)})$

$(1-2)S_k = 2 + 2^2 + 2^3 + 2^4 + \dots + 2^k - k.2^{k+1}$

$-S_k = 2(1 + 2 + 2^2 + 2^3 + \dots + 2^{(k-1)}) - k.2^{k+1}$

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

$a = 1$, $r = 2$

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

$$-S_k = \frac{2(1-2^k)}{-1} - k.2^{k+1}$$

Divide by -1

$S_k = 2(1-2^k) + k.2^{k+1}$

Recurrance Relation.

A recurrance is an equation of an equality that describes a function in terms of its value on smaller input

The recursive formula which describes recursively define sequence. is called recurrance or recurrance relation.
The information about beginning of the sequence is called intial condition.

~~Tequt~~ Techniques to Solve Recurrance Relation.

1) Characteristics equation
2) Substitution method
3) Recursion Tree Method
4) Master Method.

1. Characteristic Equation

i) Homogeneous Recurrance
ii) Inhomogeneous Recurrance
iii) Logarithmic Recurrance
iv) Range of Transformation

Π - Symbol of factors

i) Homogeous Recurrance

Homogenous Recurrance is $a_i t_{n-i} = 0$ ~~to~~ ~~o~~

$0 \le i \le k$

$a_0 t_n + a_1 t_{n-1} + a_2 t_{n-2} + a_3 t_{n-3} + \dots + a_k t_{n-k} = 0$

Replan $t_n = x^n$

$a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + a_3 x^{n-3} + \dots + a_k x^{n-k} = 0$

put $n = k$.

$a_0 x^k + a_1 x^{k-1} + a_2 x^{k-2} + a_3 x^{k-3} + \dots + a_k x^0 = 0$

— polynomial of degree k.

we have. $p(x) = a_0 x^k + a_1 x^{k-1} + a_2 x^{k-2} + \dots + a^k$

$\therefore \quad p(x) = \prod_{i=1}^{k} (x - r_i)$

$x - r_i = 0$

$x = r_i$

we have. $t_n = x^n$

$t_n = r_i^n$

i.e. $\boxed{t_n = \sum_{i=1}^{k} c_i r_i^n}$

Page No.		
Date		

for $n = 1$

$t_1 = 4c_1 - c_2$

$4c_1 - c_2 = 5$ — (b)

$c_1 = 1 \quad , \quad c_2 = -1$

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

$t_n = \theta(4^n)$

Removing constant coefficient

$4^n \quad -(-1)^n$

$\downarrow \qquad \downarrow$

$4^n \quad (-1)^n$

Q-22

★ Roots

→ Real and distinct

$$\boxed{t_n = c_1 r_1^n + c_2 r_2^n + c_3 r_3^n}$$

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

$t_n = c_1 2^n + c_2 3^n + c_3 1^n$

Solve the Recurrance Relation

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

$t_n = 3t_{n-1} + 4t_{n-2}$

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

put $t_n = x^n$

$x^n - 3x^{n-1} - 4x^{n-2} = 0$

put $n=2$

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

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

$t_n = c_1 r_1^n + c_2 r_2^n$

$= c_1 4^n + c_2(-1)^n$ — (1)

for $n=0$ in eq (1)

$t_0 = c_1 + c_2$

$c_1 + c_2 = 0$ — (2)

Page No.

Date

$3c_1 + 2c_2 = 1$ — (b)

from eq (a) and (b)

multiply eq a by -3

$-3c_1 - 3c_2 = 0$ — (c)

Adding a dnd c

$$
\begin{array}{rl}
 & \cancel{3c_1} + 2c_2 = 1 \\
+ & -\cancel{3c_1} - 3c_2 = 0 \\
\hline
 & -c_2 = 0 \\
 & c_2 = -1
\end{array}
$$

put $c_2 = -1$ in eq (a)

$\cancel{-1} + \cancel{c_2}$ $c_1 + c_2 = 0$

$c_1 - 1 = 0$

$c_1 = 1$

Removing constant Cofficient

3^n -2^n

$\downarrow$ $\downarrow$

3^n 2^n

$\therefore$ $t_n = 3^n - 2^n$

$t_n = 5t_{n-1} - 6t_{n-2}$ is in sequence ∴ It is homogeneous recurrence

1 Solve the given recurrence

$$t_n = \begin{cases} n & \text{if } n=0 \text{ or } n=1 \\ 5t_{n-1} - 6t_{n-2} & \text{otherwise} \end{cases}$$

=) $t_n = 5t_{n-1} - 6t_{n-2}$

$t_n - 5t_{n-1} + 6t_{n-2} = 0$

Put $t_n = x^n$

$x^n - 5x^{n-1} + 6x^{n-2} = 0$

put $n = 2$

$x^2 - 5x + 6 = 0$

$r_1 = 3, r_2 = 2$

$t_n = c_1 r_1^n + c_2 r_2^n$

$t_n = \cancel{c_1 r_1}\ c_1 3^n + c_2 2^n \quad — (1)$

when $n=0$, in eq (1)

$t_0 = c_1 + c_2$

$c_1 + c_2 = 0 \quad — (2)$

$n = 1$ in eq (1)

$t_1 = 3c_1 + 2c_2$

$t_n = c_1 4^n + c_2 2^n \quad -①$

for $n = 0$

$c_1 + c_2 = 1 \quad -②$

for $n = 1$

$4c_1 + 2c_2 = t_1$

$t_n = 4t_{n-1} + 2^n$

$4c_1 + 2c_2 = 6 \quad -③$

$t_1 = 4t_0 + 2$

$= 4 + 2$

$t_1 = 6$

$c_1 = 2, \quad c_2 = -1$

put the value in eq ①

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

$t_n = \theta(4^n)$

$(2)4^n \downarrow 4^n$

$(-1) 2^n \downarrow 2^n$

ii) Inhomogeneous Recurrance

when linear compination is not equal to zero then such recurrance is called inhomogeneous Recurrance

$$\underbrace{a_0 t_n + a_1 t_{n-1} + a_2 t_{n-2} + \dots + a_k t_{n-k}}_{\text{linear term}} = \underset{\text{constant}}{b^n} \cdot \underset{\text{non-linear}}{P(n)}$$

where $P(n)$ — polynomial of degree d
b — constant.

Then,

$$(a_0 x^k + a_1 x^{k-1} + a_2 x^{k-2} + \dots + a_k)(x-b_1)^{d_1+1}(x-b_2)^{d_2+1} = 0$$

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

$\Rightarrow t_n = 4t_{n-1} + 2^n$

$t_n - 4t_{n-1} = 2^n$

$x - 4 = 2^n . n^0 \Rightarrow (x-4)(x-2)^{0+1} = 0$

$(x-4)(x-2) = 0$

$r_1 = 4, \quad r_2 = 2$

$b^n \; P(n)$

$2^n . n^0$

$\therefore b = 2$

$d = 0$

$(x-1)(x^2-4x+4)=0$
$(x-1)(x-2)(x-2)=0$

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

$t_n = c_1 1^n + c_2 2^n + c_3 n 2^n$ — ①

for $n=0$
$0 = c_1 + c_2$
$c_1 + c_2 = 0$ — ②

for $n=1$
$c_1 + 2c_2 + 2c_3 = 1$ — ③

for $n=2$
$c_1 + 4c_2 + 8c_3 = 2$ — ④

$c_1 = -2, \; c_2 = 2, \; c_3 = -1/2$

put value in eq ①

$t_n = (-2)1^n + 2 \cdot 2^n - \frac{1}{2} \cdot n \cdot 2^n$

$\boxed{t_n = \Theta(n \cdot 2^n)}$

Removing constant coefficien

$(-2)1^n \xrightarrow{-2} 1^n$

$2 \cdot 2^n \xrightarrow{2} 2^n$

$-\frac{1}{2} n \cdot 2^n \xrightarrow{-1/2} n \cdot 2^n$

eg. 2) $r_1 = 2, \quad r_2 = 3, \quad r_3 = 3$

$$t_n = c_1 2^n + c_2 3^n + c_3 n 3^n$$

eg. 3) $r_1 = 2, \quad r_2 = 3, \quad r_3 = 2, \quad r_4 = 3$

$$t_n = c_1 2^n + c_2 3^n + c_3 n 2^n + c_4 n 3^n.$$

1. Solve the given recurrance

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

$\Rightarrow$ $t_n = 5t_{n-1} - 8t_{n-2} + 4t_{n-3}$

$t_n - 5t_{n-1} + 8t_{n-2} - 4t_{n-3} = 0$

put $t_n = x^n$

$x^n - 5x^{n-1} + 8x^{n-2} - 4x^{n-3} = 0$

put $n = 3$

$x^3 - 5x^2 + 8x - 4 = 0$

→ consider

$(x-1) = 0$

$x = 1$

1	1	-5	8	4
	↓	1	-4	4
	1	-4	4	$\boxed{0}$

$x^2 - 4x + 4x^2$

$$4c_1 + 8c_2 + c_3 + 2c_4 = 12 \quad -③$$

for $n = 3$

$$t_3 = 8c_1 + 24c_2 + c_3 + 3c_4$$

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

$$t_3 = 2.12 + 3 + 8$$

$$= 24 + 3 + 8$$

$$= 24 + 11$$

$$= 35$$

$$8c_1 + 24c_2 + c_3 + 3c_4 = 35 \quad -④$$

$c_1 = 2$, $c_2 = 1$, $c_3 = -2$, $c_4 = -1$

put the value eq^n ⓐ

$$t_n = 2 \cdot 2^n + 1 \cdot n \cdot 2^n - 2 \cdot 1^n - 1 \cdot n \cdot 1^n$$

$$\boxed{t_n = \theta(n \cdot 2^n)}$$

Page No.
Date

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

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

$$t_n = c_1 2^n + c_2 n 2^n + c_3 1^n + c_4 n \cdot 1^n \quad - 0$$

for $n = 0$

$$c_1 + c_3 = 0 \quad - 1$$

for $n = 1$

$$t_1 = 2c_1 + 2c_2 + c_3 + c_4$$

$$t_n = 2t_{n-1} + n + 2^n$$
$$= 0 + 1 + 2$$
$$= 3$$

$$2c_1 + 2c_2 + c_3 + c_4 = 3 \quad - 2$$

for $n = 2$

$$t_2 = 4c_1 + 8c_2 + c_3 + 2c_4$$

$$t_n = 2t_{n-1} + n + 2^n$$
$$t_2 = 2 \cdot t_1 + n + 2^n$$
$$= 2 \cdot 3 + 2 + 4$$
$$= 6 + 6 = 12$$

Page No.
Date

$t_m = 2t_{m-1} + 1$

$t_1 = 2t_0 + 1$

$t_1 = 1$

$2c_1 + c_2 = 1$ — (3)

$c_1 = 1$, $c_2 = -1$

put the value in eq (1)

$t_m = 2^m - 1^m$

$t_m = \theta(2^m)$ -- Time complexity of given recurrance

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

$b^n \cdot p(n)$

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

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

$n \downarrow 1^n \cdot n$ $\quad$ $2^n \downarrow 2^n \cdot n^0$

$t_n - 2t_{n-1} = 1^n \cdot n + 2^n \cdot n^0$

$b_1 = 1$, $d_1 = 1$, $b_2 = 2$, $d_2 = 0$

Page No.

Date

2) Solve the given recurrance.

$$t_n = \begin{cases} 0 & \text{if } m = 0 \\ 2t_{m-1} + 1 & \text{, otherwise} \end{cases}$$

$\Rightarrow$ $t_m = 2t_{m-1} + 1$ $\qquad b^n . p(n)$

$\downarrow$

$t_m - 2t_{m-1} = 1$ $\qquad 1^m . m^0$

$t_m - 2t_{m-1} = 1^m . m^0$

$b = 1$, $d = 0$

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

$r_1 = 2$, $r_2 = 1$

$t_m = c_1 2^m + c_2 1^m$ — (1)

for $m = 0$

$0 = c_1 + c_2$ — (2)

$$\cancel{4c_1} + 4c_2 + 0 + 0 = \cancel{4t_0}$$
$$- \quad \cancel{4c_1} + c_2 + 2c_3 = \cancel{4t_0} + 2$$
$$\cancel{4c_1} + c_2 + \cancel{2c_3} = \cancel{4t_0} + 4$$

$$c_2 = -2$$

$c_2 = -2$ in eq 1

$$\cancel{c_1 = c_2 - t_0}$$
$$\cancel{= -2 - t_0}$$
$$\cancel{= t_0 - 2}$$

$$t_0 = c_1 + c_2$$
$$= +c_1 - 2$$
$$t_0 + 2 = c_1$$

$t_n = c_1 2^n + c_2 1^n + c_3 n.1^n$

for $n = 0$

$t_0 = c_1 + c_2$ — (1)

for $n = 1$

$t_1 = 2c_1 + c_2 + c_3$

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

$t_1 = 2t_0 + \cancel{n}\ 1$

$2c_1 + c_2 + c_3 = 2t_0 + \cancel{n}\ 1$ — (2)

for $n = 2$

$t_2 = 4c_1 + c_2 + 2c_3$ —

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

$t_2 = 2t_1 + 2$

$= \cancel{2t_0}\ 2(2t_0 + 1) + 2$

$= 4t_0 + 2 + 2$

$= 4t_0 + 4$

$4c_1 + c_2 + 2c_3 = 4t_0 + 4$ — (3)

$$\cancel{\begin{aligned} 2c_1 + 3c_2 &= 2t_0 + 3 \\ -c_1 \pm c_2 &= -t_0 \\ \hline c_1 + 2c_2 &= t_0 + 3 \end{aligned}}$$

~~put $c_1 = 0$~~

$$\begin{aligned} \cancel{2c_1} + 3c_2 - 3 &= \cancel{2t_0} \\ \cancel{2c_1} \pm 2c_2 + 0 &= \cancel{2t_0} \\ \hline c_2 - 3 &= 0 \\ c_2 &= 3 \end{aligned}$$

$$c_1 = t_0 - 3$$

$$t_n = (t_0 - 3)\, 2^n + 3 \cdot 3^n$$
$$= \theta(3^n)$$

5) $t_n = 2t_{n-1} + n$, where $n \geq 0$

$t_n = 2t_{n-1} + n$ $\quad b^n \quad p(n)$

$t_n - 2t_{n-1} = n.$ $\quad \downarrow \quad \downarrow$

$b = 1, \; d = 1$ $\quad 1^n. \quad n$

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

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

$t_n = c_1 2^n + c_2 1^n + c_3 n \cdot 1^n.$ — (1)

4) Solve the given recurrence

$t_n = 2t_{n-1} + 3^n$, where $n \geq 0$

$\Rightarrow$ $t_n = 2t_{n-1} + 3^n$ $\qquad b^n \cdot p(n)$

$t_n - 2t_{n-1} = 3^n$ $\qquad \downarrow \quad \downarrow$

$b = 3, \; d = 0$ $\qquad 3^n \quad n^0$

$(x-2)(x-3) = 0$

$r_1 = 2, \; r_3 = 3$

$t_n = c_1 2^n + c_2 3^n$ — ⓪

for $n = 0$

$t_0 = c_1 + c_2$ — ①

for $n = 1$

$t_1 = 2c_1 + 3c_2$ — ②

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

$t_1 = 2t_0 + 3$

$2c_1 + 3c_2 = 2t_0 + 3$

23/08/22

Page No.
Date

* Master Method.

→ The Recurrance form for master method is

$\cancel{T(n)}$ $T(n) = aT(n/b) + fn$

where a and b are constant

$a \geq 1$, $b > 1$

$f(n)$ = +ve function

The time complexity ~~for~~ of such recurrance is calculated as per the given cases

case 1 :- If $f(n) = \theta(n^{\log_b a - \epsilon})$ where $\epsilon > 1$ if $\log_b^a > f(n)$

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

case 2 :- If $f(n) = \theta(n^{\log_b a})$

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

case 3 :- If $f(n) = \theta(n^{\log_b a + \epsilon})$ where $\epsilon > 1$

then $T(n) = \theta(f(n))$

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

2) $T(n) = \begin{cases} 1 \\ 4T(n/2) + n^2 & \text{If } n \text{ is exact power of 2} \end{cases}$

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

Replace n by 2^i

$T(2^i) = 4T(2^i/2) + (2^i)^2$

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

$t_i = 4t_{i-1} + 4^i$

$t_i - 4t_{i-1} = 4^i$

$(x-4)$ & $(x-4) = 0$

$r_1 = 4$, $r_2 = 4$

$t_i = c_1 4^i + c_2 \cdot i \cdot 4^i$

$n = 2^i$

$\lg n = \lg 2^i$

$\lg n = i \lg 2$

$i = \lg n$

$t_{\lg n} = c_1 4^{\lg n} + c_2 \cdot \lg n \cdot 4^{\lg n}$

$T(n) = c_1 \cdot 4^{\lg n} + c_2 \cdot \lg n \cdot 4^{\lg n}$

$$t_{\lg n} = c_1 \cdot 3^{\lg n} + c_2 \cdot 2^{\lg n}$$
$$= c_1 \cdot n^{\lg 3} + c_2 \cdot n^{\lg 2}$$

$$t_i = T(2^i)$$
$$t_{\lg n} = T(n)$$

$$T(n) = c_1 n^{\lg 3} + c_2 n \quad \text{— } (2)$$

for $n = 2^0 = 1$ (put $n=1$ in eq (1))

$$T(1) = c_1 + c_2$$
$$c_1 + c_2 = 1 \quad \text{— } (3)$$

for $n = 2^1 = 2$ (put $n = 2$ in eq^n (2))

$$T(2) = 3c_1 + 2c_2$$
$$T(n) = 3T(n/2) + n$$
$$T(2) = 3T(2/2) + 2$$
$$= 3T(1) + 2$$
$$= 3 \times 1 + 2$$
$$= 5$$

$$3c_1 + 2c_2 = 5 \quad \text{— } (4)$$

$$c_1 = 3, \quad c_2 = -2$$

$$T(n) = 3n^{\lg 3} - 2n$$
$$T(n) = \Theta(n^{\lg 3}) \quad \underline{\text{OR}}$$
$$T(n) = \Theta(3^{\lg n}$$

22/08/22

Page No.
Date

★ Logarithmic Recurrence

1) $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3T(n/2) + n, & \text{if } n \text{ is the exact po} \\ & \text{of } 2 \end{cases}$

$\Rightarrow$ $T(n) = 3T(n/2) + n$

Replace n by 2^i

$T(2^i) = 3T(2^i/2) + 2^i$

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

Replace $T(2^i)$ by t^i

Replace $T(2^{i-1})$ by t^{i-1}

$t_i = 3t_{i-1} + 2^i$

$t_i - 3t_{i-1} = 2^i$ | $b^n p(n)$ $\downarrow$

$t_i - 3t_{i-1} = 2^i \cdot i^0$ | $b^i(p(i))$

$b = 2, \; d = 0$

$(x-3)(x-2) = 0$

$r_1 = 3, \; r_2 = 2$

$t_i = c_1 3^i + c_2 3^i$ — (1)

we have,

$n = 2^i$

$\lg n = \lg 2^i$

$\lg n = i \lg 2$

$[i = \lg n]$

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

$\Rightarrow$ $a = 4$, $b = 2$

$f(n) = n$

find $n^{\log_b^a}$

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

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

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

Hence apply case 1 for master method.

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

$n^1 = \theta(n^{2-\epsilon})$

$1 = 2 - \epsilon$

$1 - 2 = -\epsilon$

$-1 = -\epsilon$

$\epsilon = 1$

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

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

$\Rightarrow$ $a = 4$, $b = 2$

$f(n) = n^2$

find $\log_b^a$

Page No.
Date

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

$\Rightarrow$ $a = 2,\ b = 2$

$f(n) = n^2$

find $n^{\log_b a}$

~~at~~ $n^{\log_b a} = n^{\log_2 2}$

$n^{\log_b a} = n$

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

Hence apply case 3 for master method

$f(n) = \Theta(n^{\log_b a + \epsilon})$

$n^2 = \Theta(n^{1+\epsilon})$

$\cancel{n^2} = \cancel{\Theta(n}\ \cancel{n^{1+\epsilon}}$

$2 = 1 + \epsilon$

$\epsilon = 1$

$T_n = \Theta(f(n))$

$= \Theta(n^2)$

Page No
Date

$T(n) = \cancel{2(1} \; 2T(n/2) + n$

$a = 2, \; b = 2$

$f(n) = n.$

find $n^{\log_b^a}$

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

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

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

Hence applying case 2 for master method

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

$\cancel{T(n)} = \cancel{\theta} = \Theta(n \cdot \lg n)$

$T(n) = \Theta(n \cdot \lg n)$

1) find time complexity for the given recurrence

$T(n) = 2T(n/2) + n^0$

$\Rightarrow$ $a = 2, \; b = 2$

$f(n) = n^0$

find $n^{\log_b a}$

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

$= n^{\lg 2} = n^{\frac{\log 2}{\log 2}} \; n^1$

$n^{\log_b a} = n$

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

Hence, applying case 1 of master method

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

$n^0 = n^{1-\epsilon}$

$0 = 1 - \epsilon$

$\epsilon = 1$

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

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

Page No.
Date

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

$a = 16, \quad b = 4$

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

apply case 2

$f(n) = \Theta(n^{\log_b a} \cdot \lg n)$
$= \Theta(n^2 \cdot \lg n)$

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

$a = 7, b = 3$

~~$n^{\log}$~~ $n^{\log_b a} = n^{\log_3 7}$
$= n^{1.77}$

apply case 3
$f(n) = \Theta(n^{\log_b a + \epsilon})$

7) $T(n) = 2T(n/2) = n3$

$\Rightarrow$ $a = 2$, $b = 2$

$n^{\log_b a} = n$

apply case 3

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

$n^3 = \theta(n^{1+\epsilon})$

$3 = 1 + \epsilon$

$\epsilon = 2$

$T(n) = \theta(n^3)$

8) $T(n) = T\left(\frac{9n}{10}\right) + n$

$a = 1$, $b = \cancel{\frac{9}{10}}$ $10/9$

$n^{\log_b a} = n^{\frac{\log 1}{\log 10/9}}$ log 1 raha to power 0

$= n^0 = 1$

~~f(n) = θ(~~

apply case 3

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

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

apply case 2

$f(n) = \theta(n^{\log_b a} \cdot \lg n)$
$= \theta(n^2 \cdot \lg n)$

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

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

$\Rightarrow$ $a=4$, $b=2$

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

apply case 3.

$f(n) = \theta(n^{\log_b a + \epsilon})$
$n^3 = \theta(n^{2+\epsilon})$
$3 = 2+\epsilon$
~~$\epsilon = 5$~~ $\epsilon = 1$

$T(n) = \theta(n^3)$

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

$\Rightarrow$ $a = 4$, $b = 2$

$f(n) = n$

find $n^{\log_b^a}$

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

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

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

Hence apply case 1 for master method.

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

$n^1 = \theta(n^{2-\epsilon})$

$1 = 2 - \epsilon$

$1 - 2 = -\epsilon$

$-1 = -\epsilon$

$\epsilon = 1$

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

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

$\Rightarrow$ $a = 4$, $b = 2$

$f(n) = n^2$

find $\log_b^a$

$lg 2 = \frac{\log 2}{\log 2}$ T(n) is using for logarithmic recurrance

8) $T(n) = 7T(n/2) + n^2$ $n^{2.81}$

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

25/08/22

Recursion Tree Method

$T(n) = 2T(n/2) + \theta(n)$ → cost of original recurrance

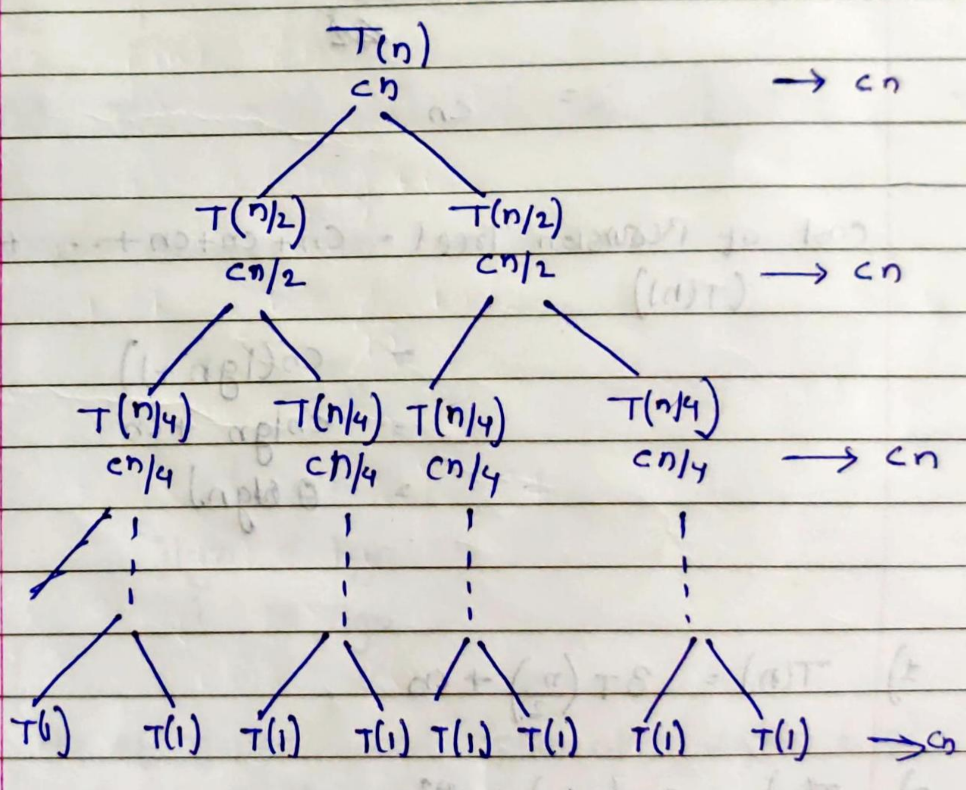
T(n) → Original Recurrance

2 → No of sub-recurrances

T(n/2) → Sub-recurrance (Sub-problem)

n/2 → Size of Sub-recurrance

Tree diagram



The subproblem at depth i

$$1 = \frac{n}{2^i}$$

$$2^i = n$$

$$i \lg 2 = \lg n$$

$$\therefore \quad i = \lg n$$

No. of nodes ~~of~~ at a perticular depth $= 2^i$

cost of node at a perticular depth $= \frac{cn}{2^i}$

Total cost of a perticular depth

$$= \cancel{2^i} \times \frac{cn}{\cancel{2^i}}$$

$$= cn$$

cost of Recursion Tree ($T(n)$) $= cn + cn + cn + \ldots + T(n)(\lg n + 1)$

$$= cn(\lg n + 1)$$

$$= cn\lg n + cn$$

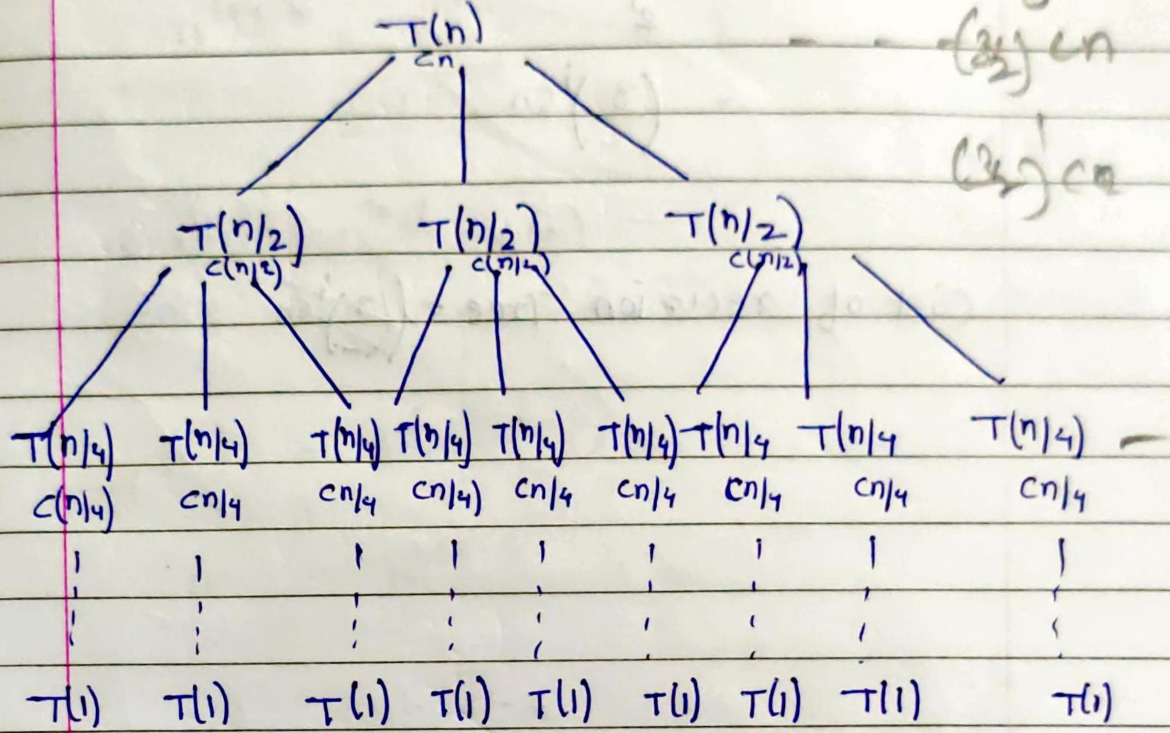
$$= \theta(n\lg n)$$

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

2) $T(n) = 2T(n/2) + cn^2$

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

- Tree diagram



- The subproblem at depth i

$$1 = \frac{n}{2^i}$$

$$2^i = n$$

$$i \lg 2 = \lg n$$

$$i = \lg n$$

No. of nodes at a perticular depth $= 3^i$

cost of node at perticular depth $= \frac{n}{2^i}$

Total cost of perticular depth

$$= 3^i \times \frac{cn}{2^i}$$

$$= \frac{3^i}{2^i} cn$$

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

Cost of recursion Tree = $\cancel{\left(\frac{3}{2}\right)^i cn}$

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

$a = 3 \quad , b = 2$

$n^{\log^a b} = n^{\log \frac{3}{2}}$

$= n^{1.58}$

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

Hence applying case 1 of master method

$f(n) = \Theta(n^{\log_b^a - \epsilon})$

$n = n^{1.58 - \epsilon}$

$1 = 1.58 - \epsilon$

$1 - 1.58 = -\epsilon$

$\epsilon = 0.58$

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

$= \Theta(n^{1.58})$

$c(n^2$

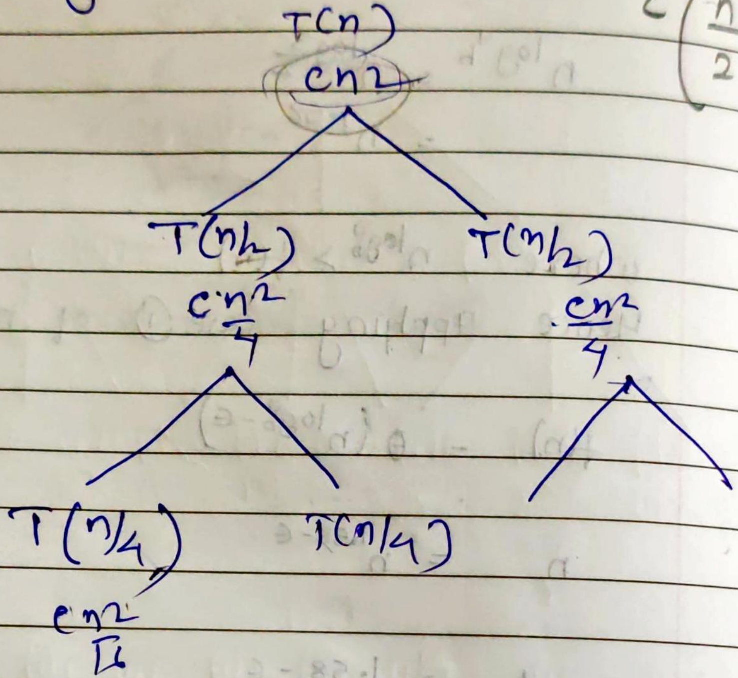
$c(n/2)^2$

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

2^i

Tree diagram

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



$$2^i \times c\left(\frac{n}{2^i}\right)^2$$

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

$$= 2^i \times c\,\frac{n^2}{4^i}$$

$$= \left(\frac{2}{4}\right)^i cn^2$$

$$= \left(\frac{1}{2}\right)^i cn^2$$

$$= \left(\frac{1}{2}\right)^0 cn^2 + \left(\frac{1}{2}\right)^1 cn^2 + \left(\frac{1}{2}\right)^2 cn^2 + \ldots + \left(\frac{1}{2}\right)^{\log n} cn^2$$

1/09/22

Page No.
Date

* Time complexity of program Segments

1) for i = 1 to n
    stm1;

The loop executed $(n+1)$ times
$$(n+1) = 1(n+1)$$
The statement will executed 1 times

$$n = 1 \times n$$

	cost	Times	Total
	1	$(n+1)$	$= 1(n+1)$
	1	n	$= 1 \times n$

$$\therefore T(n) = (n+1) + 1 \times n$$
$$= n + 1 + n$$
$$= 2n + 1$$
$$= \theta(n)$$

2) Summation method

$$T(n) = \sum_{i=1}^{n} (1)$$
$$= \sum_{i=1}^{n} 1$$
$$= 1 \times n$$
$$= n$$
$$= \theta(n)$$

$= 1 + 1 \times n$

$= 1 + n$

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

3) for $i \rightarrow 1$ to n

for $j \rightarrow 1$ to n

stm 1 ;

$$T(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} (1)$$

$$= \sum_{i=1}^{n} (n)$$

$$= n \sum_{i=1}^{n} (1)$$

$= n * 1 \times n$

$= n^2$

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

4) $l = 0$

for $i \to 1$ to n

for $j \to 1$ to n^2

for $k \to 1$ to n^3

$l = l + 1$

$$\Rightarrow \quad T(n) = 1 + \sum_{i=1}^{n} \sum_{j=1}^{n^2} \sum_{k=1}^{n^3} (1)$$

$$= 1 + \sum_{i=1}^{n} \sum_{j=1}^{n^2} (n^3)$$

$$= 1 + \sum_{i=1}^{n} n^3 \sum_{j=1}^{n^2} (1)$$

$$= 1 + \sum_{i=1}^{n} (n^3)(n^2)$$

$$= 1 + (n^3)(n^2) \sum_{i=1}^{n} (1)$$

$$= 1 + n^5 \times n$$

$$= 1 + n^6$$

$$T(n) = \Theta(n^6)$$

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

5) $\sum_{i=1}^{n} (1) = 1 \times n$

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

6) $\sum_{i=2}^{n} (1) = 1 \times (n-1)$

3) $\sum_{i=3}^{n} i = \frac{n(n+1)}{2} - 3$

7) $\sum_{i=3}^{n} (1) = 1 \times (n-2)$

4) $\sum_{i=4}^{n} i = \frac{n(n+1)}{2} - 6$

8) $\sum_{i=4}^{n} (1) = 1 \times (n-3)$

1. for i → 2 to m - 1
    for j → 3 to i
        Sum = Sum + 2 ;

$$\Rightarrow T(m) = \sum_{i=2}^{m-1} \sum_{j=3}^{i} (1)$$

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

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

$$= \frac{(m-1)(m)}{2} - 1 - 2 \left[1(m-1-1) \right]$$

Page No.
Date

$= \frac{m^2 - m}{2} - 1 - 2m + 4$

$= \frac{1}{2}(m^2 - m) - 2m + 3$

$T(m) = \Theta(m^2)$

find running time of given program segment

1) $l = 0$

for $i \to 1$ to n

for $j \to 1$ to i

for $k \to 1$ to n

$l = l + 1$

$\Rightarrow \quad T(n) = 1 + \sum_{i=1}^{n} \sum_{j=1}^{i} \sum_{k=1}^{n} (1)$

$= 1 + \sum_{i=1}^{n} \sum_{j=1}^{i} (n)$

$= 1 + \sum_{i=1}^{n} n(i)$

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

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

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

$$T(n) = \theta(n^3)$$

2) $l = 0$

for $i \Rightarrow 1$ to n

for $j = i+2$ to n

for $k = 1$ to $j-1$

$l = l+1$

$$\Rightarrow T(n) = 1 + \sum_{i=1}^{n} \sum_{j=i+2}^{n} \sum_{k=1}^{j-1} (1)$$

$$= 1 + \sum_{i=1}^{n} \sum_{j=i+2}^{n} (j-1)$$

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

$$= 1 + \sum_{i=1}^{n} \left[\cancel{\Sigma} \frac{n(n-1)}{2} - 3 - (n-2) \right]$$

$$= 1 + \cancel{\sum_{i=1}^{n}} \left[\frac{n(n-1)}{2} - 3 - (n-2) \right] \cdot \sum_{i=1}^{n} (1)$$

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

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

$$= 1 + \left[\frac{n^3+n^2}{2} - n^2 - n\right]$$

$$= 1 + \left[\frac{n^3+n^2-2n^2-2n}{2}\right]$$

$$= \cancel{1+}$$

$$T(n) = \theta(n^3)$$

3) for $i \rightarrow 1$ to $n-1$
    for $j \rightarrow i+1$ to n
        for $k \rightarrow 1$ to j
            Stmt 1 ;

$$\Rightarrow T(n) = \sum_{i\rightarrow 1}^{n-1} \sum_{j\rightarrow i+1}^{n} \sum_{k \rightarrow 1}^{j} (1)$$

$$= \sum_{i\rightarrow 1}^{n-1} \sum_{j\rightarrow i+1}^{n} (j)$$

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

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

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

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

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

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

$$T(n) = \Theta(n^3)$$

4) for i → 1 to n
   for j → n to i+1
      stmt 1;

$$\Rightarrow T(n) = \sum_{i\to 1}^{n} \sum_{j\to n}^{i+1} (1)$$

$$= \sum_{i\to 1}^{n} \cancel{(n)}\ (n-1)$$

$$= (n-1) \sum_{i\to 1}^{n} (1)$$

$$= (n-1)(n)$$

$$= n^2 - n$$

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

c) for j → 1 to n-1
for i → j+1 to n
stmt 1

→ $T(n) = \sum_{j \to 1}^{n-1} \sum_{j \to j+1}^{n} (1)$

$= \sum_{j \to 1}^{n-1} (n-1)$

$= (n-1) \sum_{j \to 1}^{n-1} (1)$

$= (n-1)(n-1)$

$= \cancel{n^2}.$

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

8/9/22

Insertion Sort

Que. Show the Snapshot of insertion sort for given array

$A = \langle 8, 6, 5, 4, 2 \rangle$

OR Illustrate the operation of insertion sort on given array $A = \langle 8, 6, 5, 4, 2 \rangle$

Sol^n Algorithm :- of instertion sort

for $j \to 2$ to length(A)
    key = A(j)
    i = j-1
    while i>0 & A(i) > key
        A (i+1) ← A(i)
        i ← i-1
    A(i+1) ← key

Execution

For j = 2 to 5
    j = 2
    key = 6
    i = 1
    while i>0 & 8>6 (T)

★ Find best & worst case running time of complexity of insertion sort.

⇒ For $j \rightarrow 2$ to length(A)
 key = A(j)
 i = j-1
 while i>0 and A(i) > key
  A(i+1) = A(i)
  i = i-1
  ~~A(i+1) = key~~
 A(i+1) = key

	cost	No. of times
for	c_1	n
key	c_2	$n-1$
i	c_3	$n-1$
while	c_4	$\sum_{j=2}^{n} t_j$
A(i+1)	c_5	$\sum_{j=2}^{n} t_j - 1$
i	c_6	$\sum_{j=2}^{n} t_j - 1$
A(i)	c_7	$n-1$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} t_j + c_5 \sum_{j=2}^{n} t_j - 1 + c_6 \sum_{j=2}^{n} t_j - 1 + c_7(n-1)$$

for best case :-
If it is already in ascending order.
$t_j = 1$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n}(1) \cancel{+ c_5 \sum_{j=2}^{n}(1)} + c_7(n-1)$$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4(n-1) + c_7(n-1)$$
$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + c_4 n - c_4 + c_7 n - c_7$$
$$= c_1 n + c_2 n + c_3 n + c_4 n + c_7 n - c_2 - c_3 - c_4 - c_7$$

Page No.
Date

Q. Illustrate operation of insertion sort on given array $A = (2,4,5,6,8)$

For $j = 2$ to 5

$j = 2$

$key = 4$

$i = 1$

while $1 > 0$ & $2 > 4$ (F)

$A(2) = 4$

For $j = 3$

$key = 5$

$i =$

while 1>0 & 5>4 (T)

A(2) = 5

i = 0

while 0>0 & A(0)>4 (F)

~~A(1)~~

A(1) = 4

4	5	6	8	2

for j = 5

key = 2

i = 4

while 4>0 & 8>2 (T)

A(5) = 8

i = 3

while 3>0 & 6>2 (T)

A(4) = 6

i = 2

while 2>0 & 5>2 (T)

A(3) = 5

i = 1

while 1>0 & 4>2 (T)

A(2) = 4

i = 0

while 0>0 & A(0)>2 (F)

A(1) = 2

2	4	5	6	8

A(2) = 8

i = 0

~~white~~ while 0>0 & A(0) > 6 (F)

A(1) = 6

6	8	5	4	2

for j = 3

key = 5

i = 2

while 2>0 & 8>5 (T)

A(3) = 8

i = 1

while 1>0 & 6>5 (T)

A(2) = 6

i = 0

while 0>0 & A(0)>5 (F)

A(1) = 5

5	6	8	4	2

~~j=4~~ .

for j = 4

key = 4

i = 3

while 3>0 & 8>4 (T)

A(4) = 8

i = 2

while 2>0 & 6>4 (T)

A(3) = 6

i = 1

$A(10) = \langle 7, 9, 15, 18, 21, 26, 31, 38, 41, 55 \rangle$

$x = 38$

$\Rightarrow \quad l = 0, \; r = 9$

Binary search $(A, 38, 0, 9)$

$m = \lfloor 9/2 \rfloor = \lfloor 4.5 \rfloor = 4$

if $38 < A(4)$

$38 \leq 21$ (F)

return Binary search $(A, x, 5, 9)$

$\boxed{26 \mid 31 \mid 38 \mid 41 \mid 55}$

Binary search $(A, 38, 5, 9)$

$m = 14/2 = 7$

if $38 \leq 38$ (T)

Return Binary search $(A, 38, 5, 7)$

$\boxed{26 \mid 31 \mid 38}$

Binary search $(A, 38, 5, 7)$

$m = 12/2 = 6$

if $38 \leq 31$ (F)

return Binary search $(A, 38, 7, 7)$ — $\boxed{38}$

Binary Search $(A, 38, 7, 7)$

if $(7 == 7)$

return 7

The element 38 is searched at position 7.

20/09/22

* Divide and Conquer strategy

1) Binary Search Algorithm
2) Strassen's matrix multiplication
3) Merge Sort
4) Quick Sort
5) Heap Sort

1 Binary Search Algorithm

Q. ⇒ Illustrate the operation of binary search to find the position of given elements for a given array. also derive its recurrance relation & find its time complexity?

⇒ Algorithm

```
Binary_search (A, x, l, r)
        if l == r
            return l
        else  m = [(l+r)/2]
            if  x ≤ A(m)  then
                return  Binary-Search (A, x, l, m)
        else
                return  Binary-Search (A, x, m+1, r)
```

$$= \frac{c_4 n^2}{2} + \frac{c_5 n^2}{2} + \frac{c_6 n^2}{2} + c_1 n + c_2 n + c_3 n$$

$$+ \frac{c_4 n}{2} + \frac{c_5 n}{2} - c_5 n + \frac{c_6 n}{2} - c_6 n + c_7 n$$

$$- c_2 - c_3 - c_4 - c_7$$

$$= n^2\left(\frac{c_4}{2} + \frac{c_5}{2} + \frac{c_6}{2}\right) + n\left(c_1 + c_2 + c_3 + \frac{c_4}{2} + \frac{c_5}{2} - c_5\right.$$

$$\left. + \frac{c_6}{2} - c_6 + c_7\right) + \left(-c_2 - c_3 - c_4 - c_7\right)$$

$= an^2 + bn + c$ (quadratic func of n)

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

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

$$= \underbrace{(c_1 + c_2 + c_3 + c_4 + c_7)}_{(a)} n + \underbrace{(-c_1 - c_2 - c_3 - c_4 - c_7)}_{(b)}$$

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

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

For worst case :

If it is ~~ch~~ ~~not~~ in descending order.

$\therefore t_j = j$

$$T(n) = c_1 n + c_2(n-1) + c_3(n-1) + c_4 \sum_{j=2}^{n} (j) + c_5 \sum_{j=2}^{n} (j-1) + c_6 \sum_{j=2}^{n} (j-1) + c_7(n-1)$$

$$= c_1 n + c_2(n-1) + c_3(n-1) + c_4\left(\frac{n(n+1)}{2} - 1\right) + c_5\left(\frac{n(n+1)}{2} - 1\right)$$

$$- c_5(n-1) + \left(c_6 \frac{n(n+1)}{2} - 1\right) - c_6(n-1) + c_7(n-1)$$

$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + c_4\left(\frac{n^2+n}{2} - 1\right) + c_5\left(\frac{n^2+n}{2} - 1\right)$$

$$- \cancel{c_5 n} - c_5 n + c_5 + c_6\left(\frac{n^2+n}{2} - 1\right) - c_6 n + c_6$$

$$+ c_7 n - c_7$$

$$= c_1 n + c_2 n - c_2 + c_3 n - c_3 + \frac{c_4 n^2}{2} + \frac{c_4 n}{2} - c_4 + \frac{c_5 n^2}{2}$$

$$+ \frac{c_5 n}{2} - \cancel{c_5} - c_5 n + \cancel{c_5} + \frac{c_6 n^2}{2} + \frac{c_6 n}{2} - \cancel{c_6}$$

$$- c_6 n + \cancel{c_6} + c_7 n - c_7$$

Page No.
Date

$P_7 = (A_{11} - A_{21})(B_{11} + B_{12})$
$= (1-7)(6+8)$
$= -6 \times 14$
$= -84$

$C_{11} = P_5 + P_4 - P_2 + P_6$
$= 48 - 10 - 8 - 12$
$= 48 - 18 - 12$
$= 48 - 30$
$= 18$

$C_{12} = P_1 + P_2$
$= 6 + 8$
$= 14$

$C_{13} = P_3 + P_4$
$= 72 + (-10)$
$= 62$

$C_{14} = P_5 + P_1 - P_3 - P_7$
$= 48 + 6 - 72 + 84$
$= 54 + 12$
$= 66$

$84 - 72 = 12$

$$\begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$$

Use stressan's algorithm to compute matrix product

$\begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix}$ Show your work. Also find recurrance relation & time complexity

$P_1 = A_{11} \cdot B_{12} - A_{11} B_{22}$

$= 1 \times 8 - 1 \times 2 \quad = 6$

$P_2 = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}$

$= 1 \times 2 + 3 \times 2 \quad = 2 + 6$

$= \cancel{8}$

$= 8$

$P_3 = A_{21} \cdot B_{11} + A_{22} \cdot B_{11}$

$= 7 \times 6 + 5 \times 6$

$= 42 + 30 \quad = 72$

$P_4 = A_{22} \cdot B_{21} - A_{22} \cdot B_{11}$

$= 5 \times 4 - 5 \times 6 \quad = 20 - 30$

$= -10$

$P_5 = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}$

$= 1 \times 6 + 1 \times 2 + 5 \times 6 + 5 \times 2$

$= 6 + 2 + 30 + 10$

$= 8 + 40 \quad = 48$

$P_6 = A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}$

$= 3 \times 4 + 3 \times 2 - 5 \times 4 - 5 \times 2$

$= 12 + 6 - 20 - 10$

$= 18 - 30$

$= -12$

$=$

$-\cancel{8}$

2. Strassen's Matrix multiplication Formulaes

$P_1 = A_{11} \cdot B_{12} - A_{11} B_{22}$

$P_2 = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}$

$P_3 = A_{21} \cdot B_{11} + A_{22} \cdot B_{11}$

$P_4 = A_{22} \cdot B_{21} - A_{22} \cdot B_{11}$

$P_5 = A_{11} \cdot B_{11} + A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}$

$= (A_{11} + A_{22})(B_{11} + B_{22})$

$P_6 = A_{12} \cdot B_{21} + A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}$

$= (A_{12} - A_{22})(B_{21} + B_{22})$

$P_7 = (A_{11} - A_{21})(B_{11} + B_{12})$

$C_{11} = P_5 + P_4 - P_2 + P_6$

$C_{12} = P_1 + P_2$

$C_{13} = P_3 + P_4$

$C_{14} = P_5 + P_1 - P_3 - P_7$

$$\underset{C}{\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}} = \underset{A}{\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}} \underset{B}{\begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}}$$

Page No. | Date

Recurrance Relation

$T(n) = aT(n/b) + \theta(D) + \theta(c)$

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

$= T(n/2) + 1$

$\boxed{T(n) = T(n/2) + 1}$

Time complexity.

$f(n) = n^0$

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

$= n^{\lg 1}$

$= n^0$

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

$= \theta(n^0 \lg n)$

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

else

$A(k) = R(j)$

$i = j+1$

⇒ Merge Sort

15	10	5	20	25	30	40	35

$p = 1$ $\quad$ $r = 8$

$q = 4$

merge-sort (A, 1, 4)
merge-sort (A, 5, 8)
merge-~~sort~~ (A, 1, 4, 8)

merge-sort (A, 1, 4)

15	10	5	20

1 $\quad$ 4

$q = 2$

merge-sort (A, 1, 2)
merge-sort (A, 3, 4)
merge-~~sort~~ (A, 1, 2, 4)

Merge Sort

~~merge-sort (A,p,r)~~

Algorithm

```
merge-sort (A,p,r)
    if p<r
        q = ⌊(p+r)/2⌋
        merge-sort (A,p,q)
        merge-sort (A,q+1,r)
        merge- sort (A,p,q,r)

merge (A,p,q,r)
    n1 = q-p+1
    n2 = r-q
    for i = 1 to n1
        l(i) = A(p+i-1)
    for j = 1 to n2
        R(j) = A(q-j)
    L(n1+1) = ∞
    R(n2+1) = ∞
    i=1, j=1
    for k = p to r
        if L(i) ≤ R(j)
            A(K) = L(i)
            i = i+1
```

Page No. Date

$P_3 = A_{21}B_{11} + A_{22}B_{11}$

$= 5\times 8 + 7\times 8$

$= 40 + \cancel{42}\,56$

$= 96$

$P_4 = A_{22}\cdot B_{21} - A_{22}B_{11}$

$= 7\times 6 - 7\times 8$

$= 42 - 56$

$= -14$

$P_5 = A_{11}B_{11} + A_{11}B_{22} + A_{22}B_{11} + A_{22}B_{22}$

$= 1\times 8 + 1\times 2 + 7\times 8 + 7\times 2$

$= 8 + 2 + 56 + 14$

$= 10 + 56 + 14 = 80$

$P_6 = \cancel{A_1}\,(A_{12} - A_{22})(B_{21} + B_{22})$

$= (3-7)(6+2)$

$= (-4)(8)$

$= -32$

$P_7 = (A_{11} - A_{21})(B_{11} + B_{12})$

$= (1-5)(8+4)$

$= (-4)(12)$

$= -48$

$C_{11} = P_5 + P_4 - P_2 + P_6$

$= 80 - 14 - 8 - 32 = 26$

$C_{12} = P_1 + P_2 = 2 + 8 = \cancel{12}\ 10$

$C_{13} = P_3 + P_4 = 96 - 14 = 82$

$C_{14} = P_5 + P_1 - P_3 - P_7 = 80 + 2 - 96 + 48 = 34$

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix} \begin{bmatrix} 8 & 4 \\ 6 & 2 \end{bmatrix} = \begin{bmatrix} 26 & 10 \\ 82 & 34 \end{bmatrix}$$

$T(n) = aT(n/b) + \theta(D) + \theta(C)$

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

$\boxed{T(n) = 7T(n/2) + n^2}$

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

$a = 7, b = 2$

$= n^{\log_b a} = n^{\log_2 7}$

$= n^{2.80}$

case 1.

$\cancel{n^{\log_b a} = n^{\log_2 7}}$

$T(n) = \theta(n^{\log_2 7} \cdot \lg n)$

$\cancel{= \theta(n^{2.80} \cdot \lg n)}$

$\boxed{T(n) = \theta(n^{2.80})}$

$$\begin{bmatrix} \overset{A_{11}}{1} & \overset{A_{12}}{3} \\ \overset{A_{21}}{5} & \overset{A_{22}}{7} \end{bmatrix} \begin{bmatrix} \overset{B_{11}}{8} & \overset{B_{12}}{4} \\ \overset{B_{21}}{6} & \overset{B_{22}}{2} \end{bmatrix}$$

$P_1 = A_{11} \cdot B_{12} - A_{11} \cdot B_{22}$

$= 1 \times 4 - 1 \times 2$

$= 4 - 2 = 2$

$P_2 = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}$

$= 1 \times 2 + 3 \times 2$

$= 2 + 6 = 8$

4. Quick sort

Show the snapshots of quick sort for given array. Also, find recurrance relation & its time complexity.

Algorithm

1) Quick sort (A, p, r)
   if p < r
      then. q = partition (A, p, r)
            Quicksort (A, p, q-1)
            Quicksort (A, q+1, r)

2) Partition (A, p, r)
      x = A(r)
      i = p - 1
      for j = p to r - 1
         if A(j) ≤ x
            then. i = i+1
                  A(i) ⟷ A(j)
      A(i+1) ⟷ A(r)

40	35

merge-sort (A, 7, 8)

q = 7

merge-sort (A, 7, 7)

merge-sort (A, 8, 8)

merge (A, 7, 7, 8)

merge-sort (A, 7, 7)

merge-sort (A, 8, 8)

merge (A, 7, 7, 8)

40	∞		35	∞

A(k) =

35	40

merge (A, 5, 6, 8)

25	30	∞		35	40	∞

A(k) =

25	30	35	40

merge (A, 1, 4, 8)

5	10	15	20	∞		25	30	35	40	∞

A(k) =

5	10	15	20	25	30	35	40

merge-~~sort~~ (A, 1, 2, 4)

10	15	∞	5	20	∞

A(k) =

5	10	15	20

merge-sort (A, 5, 8)

q = 6

merge-sort (A, 5, 6)

merge-sort (A, 7, 8)

merge (A, 5, 6, 8)

merge-sort (A, 5, 6)

25	30

q = 5

merge-sort (A, 5, 5)

merge-sort (A, 6, 6)

merge (A, 5, 5, 6)

merge-sort (A, 5, 5)

merge-sort (A, 6, 6)

merge (A, 5, 5, C)

merge-sort (A, 2, 2)
merge-sort (A, 1, 2, 2)

merge sort (A,1,1) merge-sort (A, 1, 1)
merge-sort (A, 2, 2)
merge ~~sort~~ (A, 1, 2, 2)

15	∞		10	∞

A(k) =	10	15

---

merge-sort (A, 3, 4)

5	20
3	4

q = 3
merge-sort (A, 3, 3)
merge-sort (A, 4, 4)
merge ~~sort~~ (A, 3, 3, 4)

---

merge-sort (A, 3, 3)
merge-sort (A, 4, 4)

merge ~~sort~~ (A, 3, 3, 4)

5	20

5	∞		20	∞

A(k) =	5	20

$j = 5$

$7 \le 8$ (T)

$i = 5$

$7 \leftrightarrow 7$

$j = 6$

$5 \le 8$ (T)

$i = 6$

$5 \leftrightarrow 5$

$j = 7$

$6 \le 8$ (T)

$i = 7$

$6 \leftrightarrow 6$

7	5	6	8

$q = 8$

Quicksort (A, 5, 7)

Quicksort (A, 9, 8)

q = 3
Quicksort (A, 1, 2)
Quicksort (A, 4, 3)

Quicksort (A, 1, 2) $\boxed{2}\boxed{1}$
q = partition (A, 1, 2)

partition (A, 1, 2)
i = 0
j = 1 to 1
$2 \leq 1$ (False)
$2 \longleftrightarrow 1$ $\boxed{1}\boxed{2}$

q = 1
Quicksort (A, 1, 0)
Quicksort (A, 2, 2)

Quicksort (A, 1, 0) (False)
Quicksort (A, 2, 2) $\boxed{2}$

Quicksort (A, 5, 8)
q = partition (A, 5, 8)
partition (A, 5, 8)
i = 4
for j = 5 to 7

$j = 6$

$5 \le 4$ (False)

$j = 7$

$6 \le 4$ (False)

$8 \leftrightarrow 4$

2	1	3	4	7	5	6	8

$q = 4$

Quicksort (A, 1, 3)

Quicksort (A, 5, 8)

Quicksort (A, 1, 3)

2	1	3

q = partition (A, 1, 3)

$x = 3$

$i = 0$

for $j = 1$ to 2

$j = 1$

$2 \le 3$

$i = 1$

$2 \leftrightarrow 2$

$j = 2$

$1 \le 3$ (T)

$i = 2$

$3 \leftrightarrow 3$

Date

A =

2	8	7	1	3	5	6	4

p = 1 r = 8

Quicksort (A, 1, 8)

q = partition (A, 1, 8)

partition (A, 1, 8)

i = 0

for j = 1 to 7

j = 1 $2 \leq 4$

i = 1

$2 \longleftrightarrow 2$

j = 2

$8 \leq 4$ (False)

j = 3

$7 \leq 4$ (False)

j = 4

$1 \leq 4$ (True)

i = 2

$8 \longleftrightarrow 1$

2	1	7	8	3	5	6	4

j = 5

$3 \leq 4$ (True)

i = 3

$7 \longleftrightarrow 3$

2	1	3	8	7	5	6	4

1. Fractional knapsack problem
2. Huffman code algorithm.
3. Travelling Salesman problem.
4. Minimum cost Spannig Tree.
   - krusal's algorithm
   - prim's algorithm
5. Single Source Shortest path
   - Dijkstra's algorithm.
6. Job scheduling with deadline
7. Activity selection problem

1. Frational knapsack problem

Q. Solve the given knapsack problem with all possible approaches.

where $n = 5$, $W = 100$

i	=	1	2	3	4	5
w_i	=	10	20	30	40	50
P_i	=	20	30	66	40	60
P_i/w_i	=	2	1.5	2.2	1	1.2

Approaches.

sol^n 1) Decreasing order of their profit
2) Increasing order of their weight
3) Decreasing order of ratio p_i/w_i

1) Decreasing order of their profit

Job. No.	Weight	profit	Remaining capacity of knapsack
3	30	66	100-30 = 70
5	50	60	70-50 = 20
4	20	20	20-20 = 0
		146	

2) Increasing order of their weight

Job. No.	Weight	profit	Remaining capacity of knapsack.
1	10	20	100-10 = 90
2	20	30	90-20 = 70
3	30	66	70-30 = 40
4	40	40	40-40 = 0
		156	

p_i w_i
50 60
40 ×

b_i p_i
7 7
4 × 2

3. Decreasing order of P_i/w_i

P_i/w_i	Job No.	weight	profit	Remaining capacity of knapsack
2.2	3	30	66	100 - 30 = 70
2	1	10	20	70 - 10 = 60
1.5	2	20	30	60 - 20 = 40
1.2	5	40	48	40 - 40 = 0
			164	

Q. $n = 7$, $W = 15$

i =	1	2	3	4	5	6	7
w_i =	2	8	5	7	1	4	1
p_i =	10	5	15	7	6	18	3
p_i/w_i =	5	1.6	3	1	6	4.5	3

1. Decreasing order of their ~~weight~~ profit.

Job No.	weight	profit	Remaining capacity of knapsack
6	4	18	15 - 4 = 11
3	5	15	11 - 5 = 6
1	2	10	6 - 2 = 4
4	4	4	4 - 4 = 0
		47	

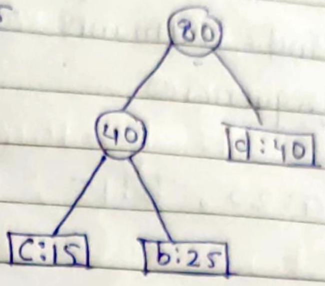
Increasing order of their weight

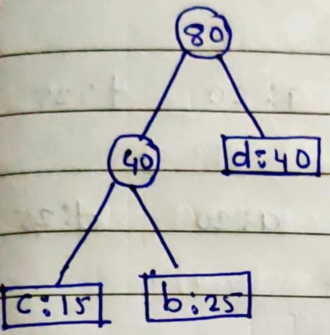
Job No.	weight	profit	Remaining capacity of knapsack
5	1	6	15-1 = 14
7	1	3	14-1 = 13
1	2	10	13-2 = 11
2	3	5	11-3 = 8
6	4	18	8-4 = 4
9	4	12	4-4 = 0
		54	

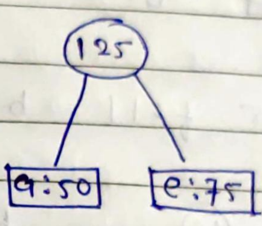
Decreasing order of Pi/wi

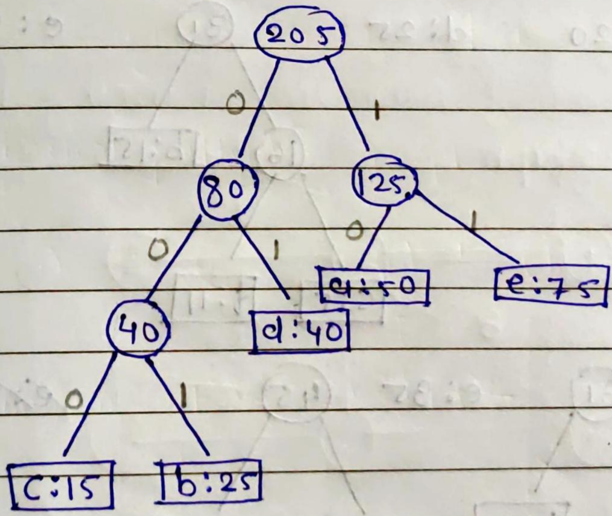
Job NO.	weight	profit	Remaining capacity of knapsack
~~1~~	~~2~~	~~10~~	~~15-2 = 13~~
~~6~~	~~4~~	~~18~~	~~13-4 = 9~~
5	1	6	15-1 = 14
1	2	10	14-2 = 12
6	4	18	12-4 = 8
3	5	15	8-5 = 3
7	1	3	3-1 = 2
2	2	3.3	2-2 = 0
		55.3	

a : 50 e : 75









a : 50 = 10

b : 25 = 001

c : 15 = 000

d : 40 = 01

e : 75 = 11

2. Huffman Code Algorithm

Algorithm

Huffman (c)

$n \leftarrow |c|$

$Q \leftarrow c$

for $i \leftarrow 1$ to $n-1$

  z.left = $x \leftarrow$ extract-min (Q)

  z.right = $y \leftarrow$ extract-min (Q)

  z.freq = x.freq + y.freq

  INSERT (Q, z)

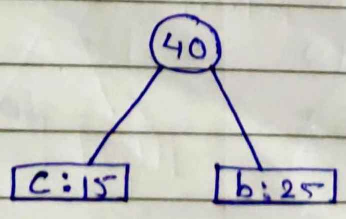
Return extract-min (Q)

Que. 1 Find Hufmann codes for following set of frequencies

~~a: 25, b: 80~~ a: 50, b: 25, c: 15, d: 40, e: 75

(1) c: 15 b: 25 d: 40 a: 50 e: 75

(2)



d: 40 a: 50 e: 75

shortest path

$H_1 \overset{5}{\text{———}} H_2 \overset{2}{\text{———}} H_3 \overset{1}{\text{———}} H_4 \overset{7}{\text{———}} H_5 \overset{6}{\text{———}} H_1$

Minimum Travel cost = 5+2+1+7+6

= 21

Qu. 2

	a	b	c	d
a	0	5	$\boxed{2}$	3
b	2	0	4	$\boxed{1}$
c	6	$\boxed{3}$	0	5
d	$\boxed{2}$	1	6	0

⇒ shortest path

$a \overset{2}{\text{———}} c \overset{3}{\text{———}} b \overset{1}{\text{———}} d \overset{2}{\text{———}} a$

Minimum cost = 2+3+1+2

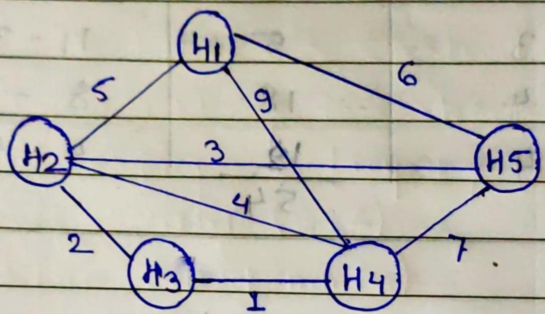
= 8

11/10/22

Page No.
Date

3. Travelling Salesman Problem.

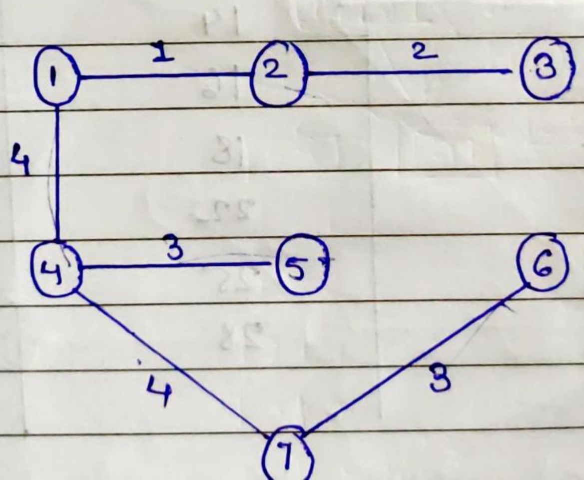
Q1. A newspaper agent daily drops the newspaper to the area asigned in such a way that he/she has to cover all the houses in the respective area. with minimum travel cost find minimum travel cost.



⇒

	H1	H2 ✓	H3 ✓	H4 ✓	H5 ✓
H1	0	$\boxed{5}$	∞	9	6
H2	5	0	$\boxed{2}$	4	3
H3	∞	2	0	$\boxed{1}$	∞
H4	9	4	1	0	$\boxed{7}$
H5	$\boxed{6}$	3	∞	7	0

Edges in ascending order of their weight	weight of Edges	Include / Exclude
{1, 2}	1	✓
{2, 3}	2	✓
{4, 5}	3	✓
{6, 7}	3	✓
{1, 4}	4	✓
{2, 5}	4	✗
{4, 7}	4	✓
{3, 5}	5	✗
{2, 4}	6	✗
{3, 6}	6	✗
{5, 7}	7	✗
{5, 6}	8	✗



4. Minimum cost Spanning Tree

1) kruskal's algorithm

Mst-kruskal (G, w)

A ← ∅

for each vertex v ∈ V(G)

make-set (v)

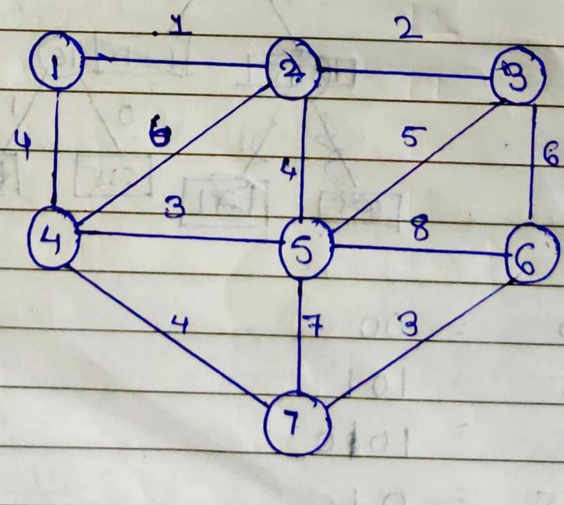
for each edge (u, v) ∈ E(G)

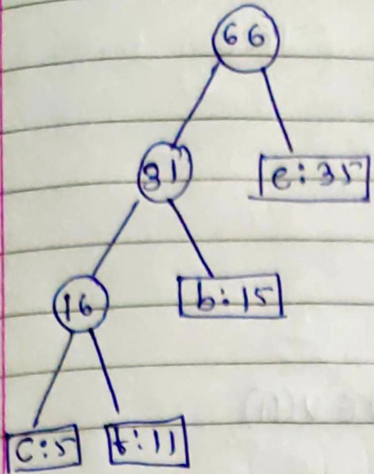
if find-set (u) ≠ find-set (v)

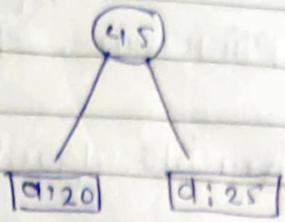
A = A + w(u, v)

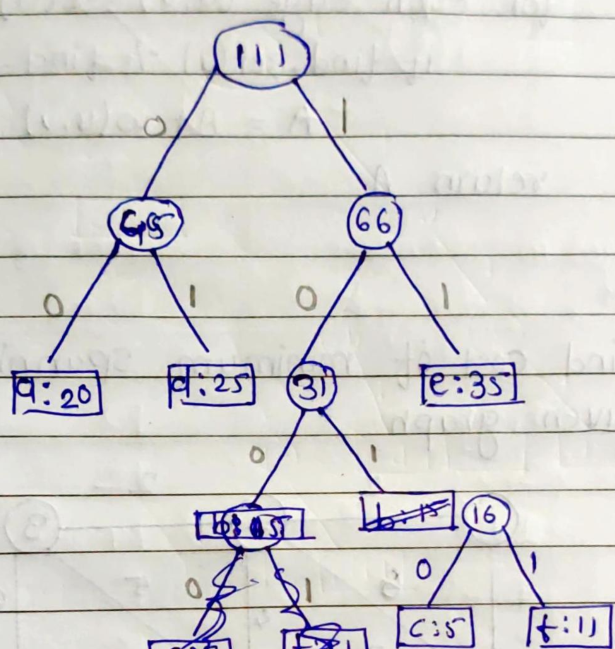
return A

Qu.1 Find cost of minimum spanning tree for given graph









a : 20 = 00

b : 15 = 101

c : 5 = 1010

d : 25 = 01

e : 35 = 11

f : 11 = 1011

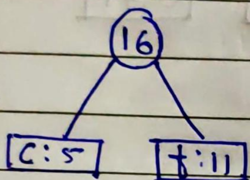
Que. 2 Find Hufmann codes for the given set of frequencies.

a : 20 , b : 15 ; c : 5 , d : 25 , e : 35 f : 11

→ ~~f : 11~~ b :

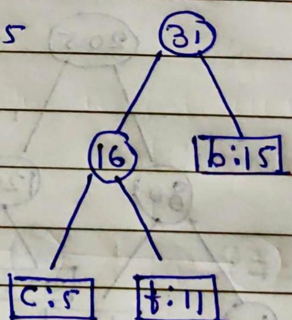
① c : 5 f : 11 b : 15 a : 20 d : 25 e : 35

2



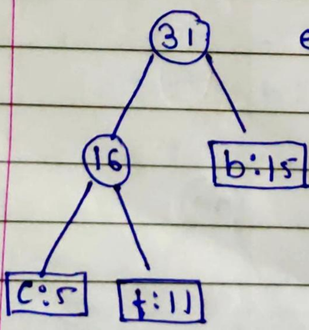
b : 15 a : 20 d : 25 e : 35

3 a : 20 d : 25

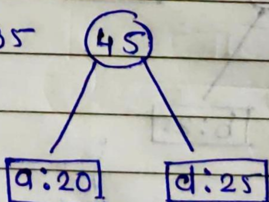


e : 35

4



e : 35



~~e : 35~~

2) Prims Algorithm

MST-prims (G, w, r)

for each $u \in V(G)$

  key $(u) = \infty$

  $\pi(u) = NIL$

key $(r) = 0$

$Q \leftarrow V(G)$

while $Q \neq \phi$

  $u \leftarrow$ extract-min (Q)

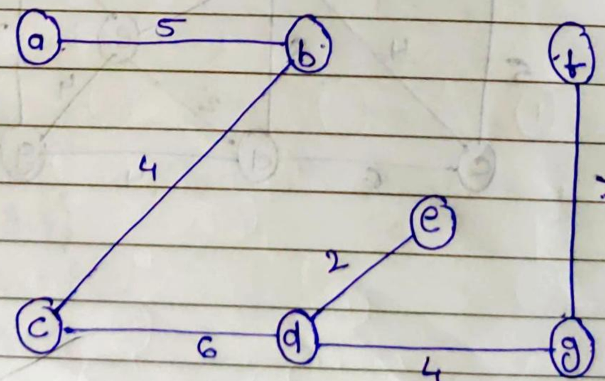
  for each $v \in adj(u)$

    If $v \in Q$ & $w(u, v) <$ key(v)

      $\pi(v) = W$

      key $(v) = w(u, v)$

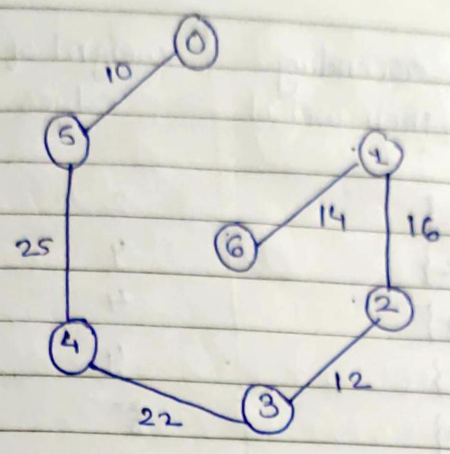
Edges in ascending order of their weight	weight of edges	Include / Exclude
{d, e}	2	✓
{b, c}	4	✓
{d, g}	4	✓
{e, g}	4	✗
{a, b}	5	✓
{a, c}	5	✗
{c, d}	6	✓
{b, e}	6	✗
{g, f}	7	✓
{b, f}	8	✗
{e, f}	9	✗
{b, d}	8	✗



Minimum cost spanning tree

$= 5 + 4 + 6 + 4 + 2 + 7$

$= 28$

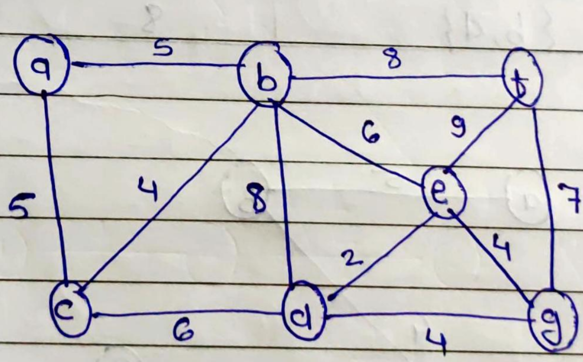


cost of minimum Spanning tree

$= 10 + 25 + 22 + 12 + 16 + 14$

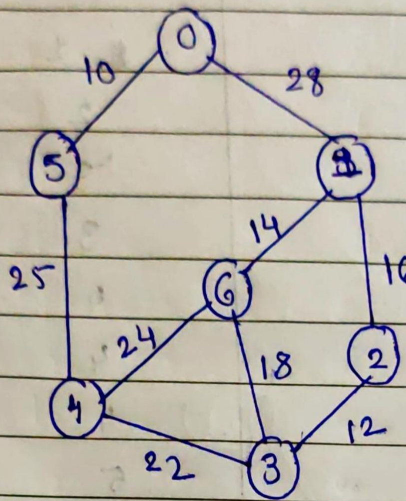
$= 99$

3)



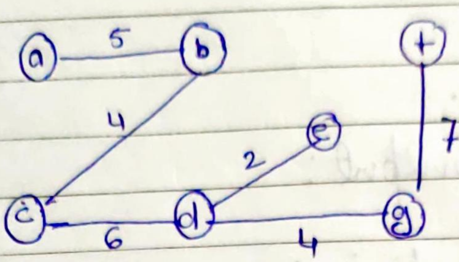
Page No. | Date

Qu. 2) Compute cost of Spanning tree for given graph

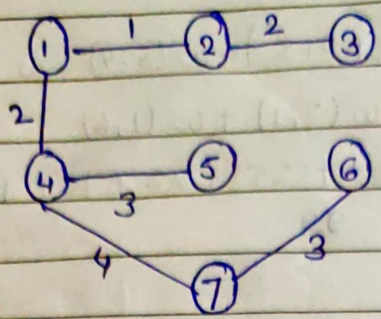


Edges in ascending order of weight	weight of Edges	include / Exclude
{0,5}	10	✓
{2,3}	12	✓
{1,6}	14	✓
{1,2}	16	✓
{3,6}	18	✗
{3,4}	22	✓
{4,5}	25	✓
{0,1}	28	✗

Page No.
Date



Cost of MST $= \omega(a,b) + \omega(b,c) + \omega(c,d) + \omega(d,e)$
$+ \omega(d,g) + \omega + (g,f)$
$= 5 + 4 + 6 + 2 + 4 + 7$
$= 28$

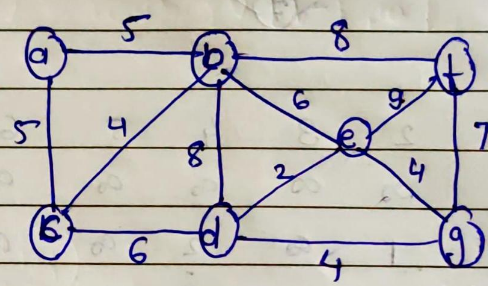


cost of MST $= \omega(1,2) + \omega(2,3) + \omega(1,4) + \omega(4,5) + \omega(4,7) + \omega(7,6)$

$= 1 + 2 + 2 + 3 + 4 + 3$

$= 15$

Q. 3

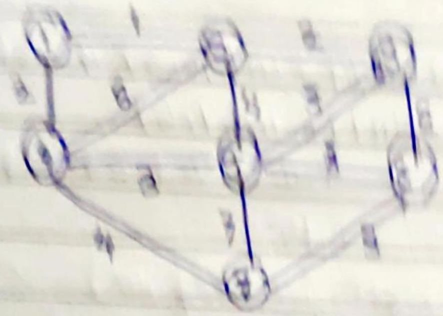


Q →	a	b	c	d	e	f	g
Key →	0	∞	∞	∞	∞	∞	∞
	-	5	5	∞	∞	∞	∞
	-	-	4	8	6	8	∞
	-	-	-	6	6	8	∞
	-	-	-	-	2	8	4
	-	-	-	-	-	8	4
	-	-	-	-	-	8	-
	-	-	-	-	-	7	-
	-	-	-	-	-	-	-

Cost of MST = [illegible]

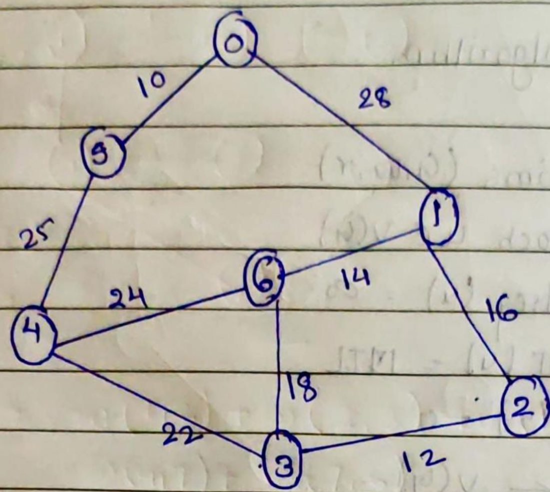
= [illegible]

= [illegible]



Q →	1	2	3	4	5	6	7
Key →	0	∞	∞	∞	∞	∞	∞
	=	1	∞	3	∞	∞	∞
	=	=	3	[illegible]	4	∞	∞
	=	=	7	3	4	6	∞
	=	=	=	=	5	6	4
	=	=	=	=	=	6	4
	=	=	=	=	=	6	=
	=	=	=	=	=	=	=

find the mst for indirected graph given below where $r=0$



Q →	0	1	2	3	4	5	6
key →	0	∞	∞	∞	∞	∞	∞
	-	28	∞	∞	∞	10	∞
	-	28	∞	∞	25	-	∞
	-	28	∞	22	-	-	24
	-	28	12	-	-	-	18
	-	16	-	-	-	-	18
	-	-	-	-	-	-	14
	-	-	-	-	-	-	-