

UNIVERSITY

Name : Sadman
..... 0112215922 School/College
Class : Section :
Roll No: Year: 1st
Subject: DSA 1

6/2
27/5/23

DSA 1

Data structure:

→ Array

→ Linked list

→ Stack

→ Queue

→ Tree

→ Graph

→ Linear

→ Non-linear

$$A = C$$

$$B = D$$

$$A + D = 180$$

$$\Rightarrow a + (a + 30) = 180$$

$$\Rightarrow 2a + 30 = 180$$

$$\Rightarrow a = 75$$

Q6 Insertion sort?

9	5	1	4	3
---	---	---	---	---



5	9	1	4	3
---	---	---	---	---



1	5	9	4	3
---	---	---	---	---

11.0

Algorithm: \rightarrow

~~Cost~~

for $i \leftarrow 2$ to n do $C_2 = 0$

~~key~~ $\leftarrow A[i]$ $C_1 = 0$

insert $A[i]$ into ~~sorted~~ $C_3 = 0$

$A[1 \dots i-1] = C_3 = 0$

$i \leftarrow i-1$ C_4 \leftarrow find $A[i] > key$ C_5
while ($i > 0$ and $A[i] > key$)

$A[i+1] \leftarrow A[i]$ C_6

$i = i-1$ C_7

$A[i+1] = key$ C_8

$[\text{ }]$

Time complexity

$$C_1 \rightarrow n - 1$$

$$C_2 \rightarrow n - 1$$

$$C_3 \rightarrow n - 1$$

$$C_4 \rightarrow n - 1$$

$$C_5 \rightarrow \frac{C_5(n-1)}{2} / \frac{C_5(n-1)w}{2}$$

$$C_6 \rightarrow \cancel{0} / \frac{C_6(n-1)w}{2}$$

$$C_7 \rightarrow 0 / \frac{C_7(n-1)w}{2}$$

$$C_8 \rightarrow n - 1$$

Case : I (Best)

1	2	3	6
---	---	---	---

Comp : 2
move : 0

case : 2 (Avg)

2	2	3	6	4
---	---	---	---	---

Comp : 2
move : 2

case : 3 (Worst)

7	2	3	6	0
---	---	---	---	---

Comp : 9
move : 9

comparison	$J=2$	$J=3$	$J=4$	\dots	$J=n$
min $C_5 \cancel{\times} (n-1)$	2	2	2	\dots	2

max $C_5 \cancel{\times} (n-1)n$	2	2	3	\dots	$n-1$
max	2	2	3	\dots	$n-1$

max	0	0	0	\dots	0
max	0	0	0	\dots	0

max $C_6(n-1)n$	2	2	3	\dots	$n-1$
max	2	2	3	\dots	$n-1$

max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$
max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$

max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$
max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$

max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$
max	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$	$n(n-1)$

0.5.23

DSA 2

Best Case: $\Theta(1)$ when $n = 1$

$$c_1 n + c_2 (n-1) + 0 + c_4 (n-1) + c_5 (n-1)$$

$$+ 0 + 0 + c_8 (n-1)$$

$$\Rightarrow (c_1 + c_2 + c_4 + c_5 + c_8) n + (-c_2 - c_4 - c_5 - c_8)$$

$f(n)$

$$\Rightarrow An + B$$

Worst Case:

$$c_1 n + c_2 (n-1) + 0 + c_4 (n-1) + c_5 \frac{n(n-1)}{2}$$

$$+ c_6 \frac{n(n-1)}{2} + c_7 \frac{n(n-1)}{2} + c_8 (n-1)$$

$$f(n) \Rightarrow An^v + Bn^l + C$$

$$\Rightarrow c_1 n^v + c_2 n^l - c_2 + 0 + c_4 n^v - c_4 + \frac{1}{2} c_5 n^v -$$

$$+ \left[c_6 n^l + \frac{1}{2} c_5 n^v + \frac{1}{2} c_6 n^l \right] + c_7 n^l + \frac{1}{2} c_5 n^v -$$

$$\frac{1}{2} c_6 n^l + \frac{1}{2} c_7 n^l + \frac{1}{2} c_7 n^v + c_8 n^v - c_8$$

$$+ \frac{1}{2} c_5 - \frac{1}{2} c_6 - \frac{1}{2}$$

$$\Rightarrow \left(\frac{1}{2} c_5 + \frac{1}{2} c_6 + \frac{1}{2} c_7 \right) n^v + (c_1 + c_2 + c_4 + c_8) n^l$$

$$\Rightarrow (-c_2 - c_4 - c_8)$$

$$\text{Best: } 1+1+n+n-1+n-1+1+n-1+n-1$$

$$\Rightarrow 5n + 3 - 4 \Rightarrow 5n - 1$$

$n \leftarrow \text{Length}[A]$ $\frac{1}{n}$
 $m \leftarrow \text{Length}[B]$ $\frac{1}{m}$
 $\text{for } i \leftarrow 1 \text{ to } n-1 \text{ do } (n)$
 $\quad \min \leftarrow i \quad (n-i)$
 $\quad \text{for } j \leftarrow 1 \text{ to } m \text{ do } (n-i) / (m-j)$
 $\quad \quad \text{if } ((A[i] + B[j]) > 10) \text{ then}$
 $\quad \quad \quad \text{break } (n-i) \rightarrow \text{break } (n-i) \quad (n-i) \quad (m-j)$
 $\quad \quad \quad \text{print } A[\min] \quad (n-i)$

~~$\Rightarrow 1 + w + n-1 + n-1 + \dots + n-2 +$~~
 ~~$w + n-1 + n-1 + \dots + n-2 +$~~

~~$\Rightarrow 1 + w + n-1 + n-1 + \dots + n-2 +$~~
 $\Rightarrow 3 - 5w - 4$

(Worst)

 ~~$\Rightarrow w + n-1 + n-1 + \dots + n-2 +$~~
 $\Rightarrow w + n-1 + n-1 + \dots + n-2 +$
 $\Rightarrow \{ (n-1)(m-1) \} + 0$

$\Rightarrow h + n-1 + n-1 + \dots + n-2 +$

$\Rightarrow w + 2nm + m + 2 + m + 1 + 0$

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

~~$$j = 1 + j = 2 + ? - -$$~~

~~for $i \leftarrow 1$ to n do $j \leftarrow 1$~~

~~for $j \leftarrow 1$ to i do $w \leftarrow w + i$~~

~~print (i, j) $w \leftarrow w + (n-i)$~~

\Rightarrow Best: ~~$n+1 + n + n+1 \rightarrow 3n + 2$~~

~~Worst:~~

~~$n+1 + nm + nm + n + m - 1$~~

~~$\Rightarrow 2nm + m - 1$~~

$$\rightarrow 1 + 1 + n + n - x + nm - m + hm - n - m + x + x + n - x$$

$$\Rightarrow 2nm + n - 2m + 2$$

P.T.O.

$$10^8 n^{x+1}$$

3

for ($i=2, i \leq w; i \leftarrow 2$) {

 print $\<(4/d)^w \rightarrow i)$ $\rightarrow 10^8 n^w$.

} // for ($i=2$) loop

Value: $\log_2 n + 1$ $\rightarrow n = 2^k$ $\rightarrow k = \log_2 n$

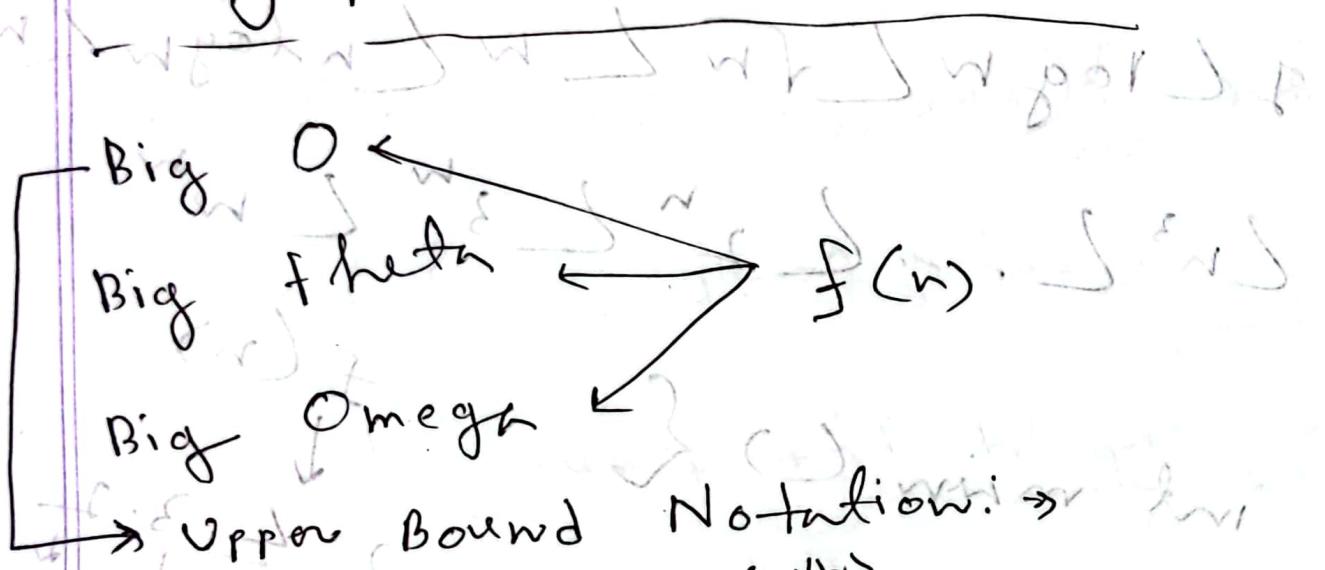
2	$0 \rightarrow 2$
4	$1 \rightarrow 2^2$
8	$2 \rightarrow 2^3$
16	$3 \rightarrow 2^4$

$$T(n) \rightarrow \log_2 n + 1 + \log_2 n$$

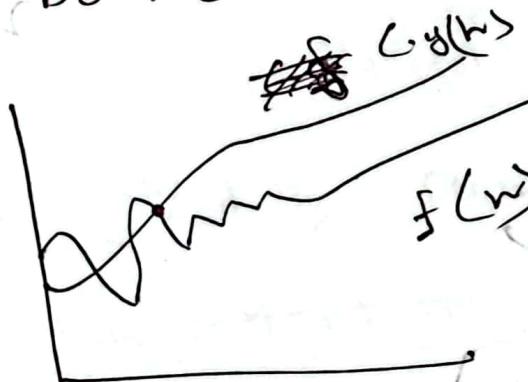
$$\Rightarrow 2 \log_2 n + 2 + \text{constant} <$$

15.0

Asymptotic Notation \rightarrow



Upper Bound Notation: $f(n) = O(g(n))$



$a_n + b$

$$f(n) \leq c \cdot g(n)$$

$f(n) = O(g(n)) \quad [n \geq 1]$

$$2^{n^v} + 5n + 3 \leq 2^{n^v} + 5n^v + 3n^v$$

$$2^{n^v} + 5n + 3 \leq 10n^v$$

$$\downarrow \quad \downarrow \\ g(n)$$

$$f(n) = O(n^v)$$

* without loss of generality

$\log w \leq \sqrt{w} \leq w \leq \log w^{\frac{1}{n}}$

$\log^3 w \leq \sqrt[3]{w} \leq w^{\frac{1}{3}}$

$f(n) = \int_{\min(1, n)}^{\max(1, n)} g(x) dx$

$a = 10$

$O(1)$

lower bound

$$f(n) \geq \int_n^{2n} g(x) dx$$

$$\Rightarrow (2n+3) \geq \int_n^{2n} g(x) dx [n=2]$$

and $n^2 + n^2 \geq g(x)$

$$(n^2)^2 \geq g(x)$$

$$(n^2)^2 > (n^2)^2$$

~~6W
3.6.23~~

DSA2

$$\text{sum} = 0 + \frac{1}{2} + h(n+1)$$

for ($i=1; i \leq n; i++$) { $n+1$

for ($j=1; j \leq 2; j++$) { $\frac{n(n+1)}{2}$

$$\text{sum} = \text{sum} + j + \frac{n(n+1)}{2} - 1$$

} $\Rightarrow \text{sum} = \text{sum} + j$

print f("%d", sum); n

$$\rightarrow \cancel{4w+4+n+h+n+w-2}$$

$$\rightarrow \frac{6w+2n+2}{2}$$

$$\rightarrow 1 + h + 1 + \frac{n+n}{2} + \frac{w+w-2}{2} + w$$

$$\rightarrow \cancel{2+2w+2+n+w+n+w-2+n}$$

$$\rightarrow \cancel{5w+2n+2} \Rightarrow$$

$$\rightarrow \cancel{2w+2} + \frac{n+n}{2} + \frac{w+w-2}{2}$$

⇒

$$\Rightarrow 2w + 2 + \frac{h^v + h}{2} + \frac{h^v + w - 2}{2}$$

~~then $(h+1)^v + (w-1)^v = 2^v$~~

$$6h^v + 2w^v + 2$$

$$\Rightarrow \frac{6h^v + 2w^v + 2}{2} \leq 2^v + 2^v + 2^v$$

$$\Rightarrow \frac{h^v + 3w^v + 2}{5} \leq 2^v + 2^v + 2^v$$

$$h^v + 3w^v + 2 \leq h^v + 3h^v + h^v$$

$$h^v + 3w^v + 2 \leq 5h^v$$

$\left\{ \begin{array}{l} h=5 \\ w=2 \end{array} \right.$

Row major order:

0	1	2	3	4	5	6	7	8	9	10

$a[0][0] [0][1] [0][2] [0][3] [1][0] [1][1] [1][2] [1][3]$

$= [0][0] [1][0] [2][0] [3][0] [0][1] [1][1] [2][1] [3][1]$

$\frac{5-w^v}{5} + \frac{w^v}{5} + 2^v + 2^v + 2^v$

add $a[i]$ to 2D Array

Elements which needs to be added in each row

2D array $i \rightarrow$

$a[i][j] = b + (i-l_1) \times (\text{column size}) \times$
component length + $(j-l_2) \times$

component
length

$$= b + (i-l_1) \times (l_2 - l_1 + 1) \times L + (j-l_2) \times L$$

0	1	2	3	4	5	6	7	8	9

$3 - 0 + 1$

$+ 4 + 00$

P.T.O

0	1		
1	2		

(00) [01] [02]
 [10] [11] [12]

① Matrix $[2][3]$ (4x2s) bbo +
 $b = 100$, size of each element is 4
 bytes.

find the memory address for
 matrix $[1,2]$ for row major
 order

$$\Rightarrow 100 + (1-0) \times (2-0+1) \times 4 +$$

$$(2-0) \times 4$$

$$\Rightarrow 100 + 12 + 8 = 120$$

matrix $\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$

$b = 100$, size $11 \times 6 \times 10$

memory add $\begin{bmatrix} 2, 1, 4 \end{bmatrix}$

$$\begin{bmatrix} 8 & 1 & 1 & 8 & 1 \end{bmatrix} = A$$

$a[i, j, k]$

$$= b + (j - l_1) \times (v_2 - l_2 + 1) \times (v_3 - l_3 + 1) \times \\ + (j - l_2) \times (v_3 - l_3 + 1) + (k - l_3) \times L$$

$$\begin{bmatrix} 8 & 1 & 8 & 0 \end{bmatrix} = 0$$

Linear Search

→ linear

$$\begin{array}{c} \text{Search} \Rightarrow \bigcirc \{n\} \\ \begin{bmatrix} 2 & 3 & 8 & 0 \\ a & a & c & 1 & 1 \end{bmatrix} = 8 \\ \text{→ } I.0 \end{array}$$

Binary Search: insertion

3 5 7 → P 11 13 17 search

Counting Sort

A =

1	3	1	2	3
0	1	2	3	

 Index

C =

0	2	1	2
0	1	2	3

 Frequency Array

C =

0	2	3	5
0	2	3	5

 Cumulative Sum Array

B =

1	1	2	3	3
1	1	2	3	3

 Sorted Array

iteration - 1

0	1	2	3
0	2	3	4

iteration - 2

0	1	2	3
0	2	2	4

iteration - 3

0	1	2	3
0	1	2	4

IT - 4

0	1	2	3
0	1	2	3

IT - 5

0	1	2	3
0	0	2	3

	1	2	3	4	5	6	7	8	9	10	
A =	2	1	4	1	3	2	7	9	2	1	

	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	2	2	2	2	0	0	1	0	2	0	1

	0	1	2	3	4	5	6	7	8	9	10	11
L =	0	2	4	5	6	6	6	7	7	8	9	10

	7	8	2	3	4	5	6	7	8	9	10	
B =	1	2	2	2	3	6	7	9	7	1	0	11

	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	1	2	4	5	6	6	6	7	7	9	10

	0	1	2	3	4	5	6	7	8	9	10	11
L =	0	2	3	5	6	6	6	7	7	9	9	10

3	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	1	3	5	6	6	6	7	7	8	9	10

4	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	1	3	5	6	6	6	6	7	8	9	10

5	0	1	2	3	4	5	6	6	6	7	18	9
C =	0	1	2	5	6	6	6	6	7	18	9	10

6	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	1	2	4	6	6	6	6	7	8	9	10

7	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	0	2	4	16	6	6	6	7	8	9	10

8	0	1	2	3	4	5	6	7	8	9	10	11
C =	0	0	2	4	5	6	6	6	7	8	9	10

9	0	1	2	3	4	5	6	6	7	7	8	9
C =	0	0	2	4	5	6	6	7	7	8	9	10

10	0	1	2	3	4	5	6	6	7	7	8	9
C =	0	0	2	4	5	6	6	7	7	8	9	9

Counting sort

A = ~~1 2 3 4 5~~

1	2	3	4	5
1	2	3	4	5

C = ~~0 2 2 1 1~~

0	1	2	3	4
0	2	2	1	1

C = ~~0 2 3 4 5~~

0	2	3	4	5
0	2	3	4	5

B = ~~2 2 2 3 4 5~~

2	2	2	3	4	5
2	2	2	3	4	5

I+3

C = ~~0 2 3 4 4~~

0	2	3	4	4
0	2	3	4	4

C = ~~0 2 3 3 4~~

0	2	3	3	4
0	2	3	3	4

C = ~~0 2 2 3 4~~

0	2	2	3	4
0	2	2	3	4

6102 spw09

3

0	1	2	3	4
0	7	2	3	4

4

0	1	2	3	4
0	0	2	3	4
11	81	F		

02
02

02
02

11
11

81
F

02	02	02
02	02	02

81	11	F
81	11	F

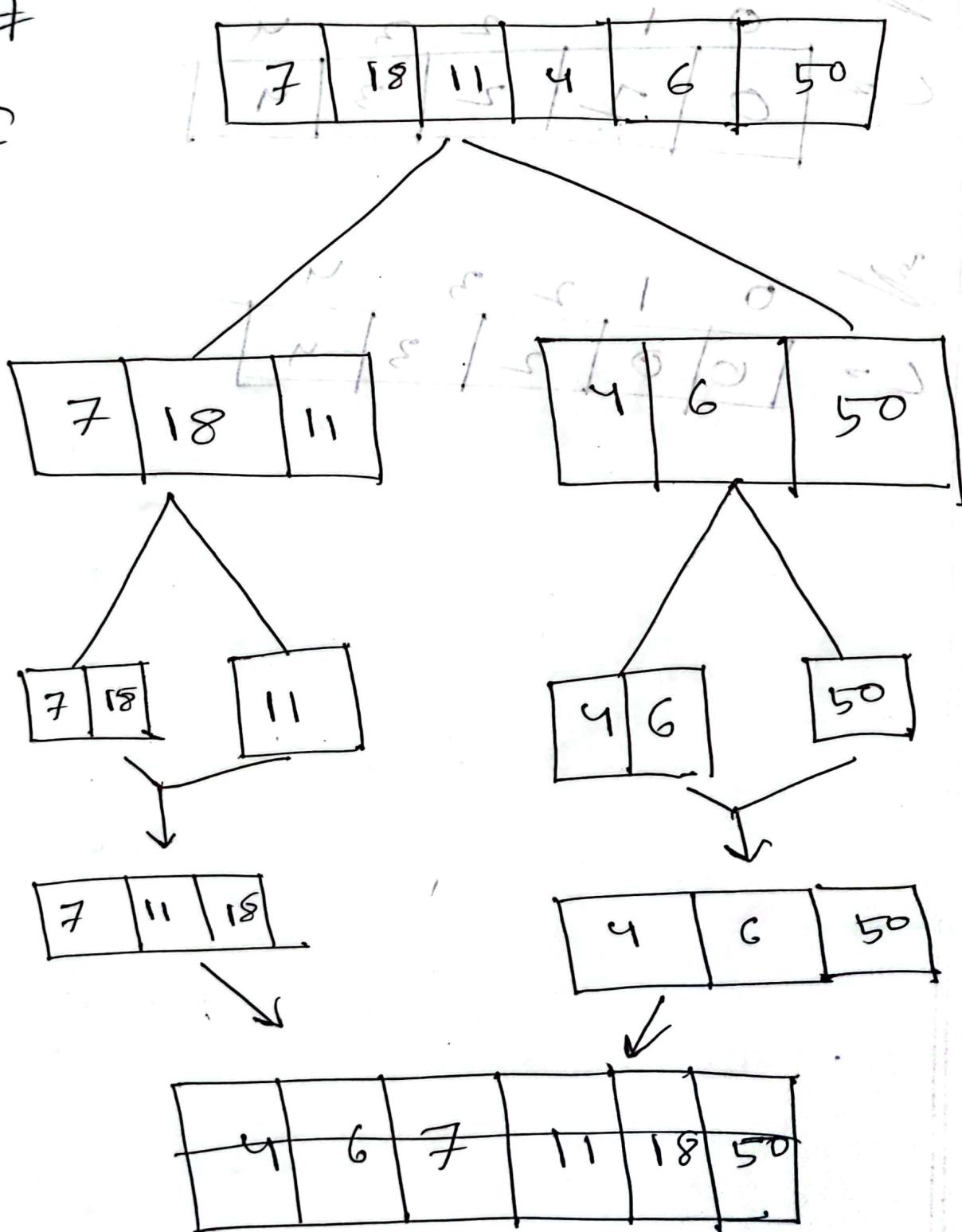
02	81	11	F	3	P
02	81	11	F	3	P

6W
10.6.23

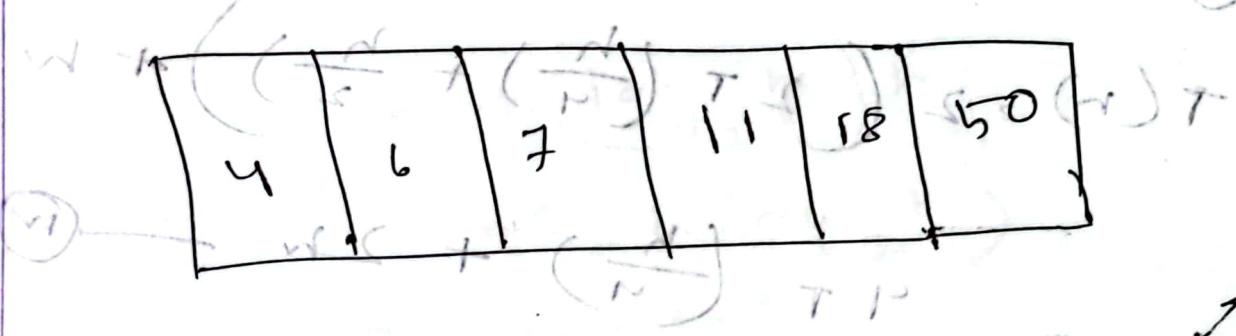
DSA

Merge sort

(1)



(1)



Time complexity:

Substitution method:

$$T_n = \begin{cases} 1 & \text{if } n=1 \\ 1 + T\left(\frac{n}{2}\right) + O(n) & \text{if } n > 1 \end{cases}$$

$$\Rightarrow 2T\left(\frac{n}{2}\right) + n \quad \text{if } n > 1$$

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

$$\Rightarrow T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \frac{n}{4}$$

1.5^n

⑪ → ①

$$T(n) \geq 2 \left[T\left(\frac{n}{4}\right) + \frac{n}{2} \right] + w$$
$$\Rightarrow 4T\left(\frac{n}{4}\right) + 2w \rightarrow ⑫$$

⑪ → ⑭

$$→ 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2w$$

$$= 8T\left(\frac{n}{8}\right) + 3w$$

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

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

$$⑭ = T\left(\frac{n}{2^k}\right) \leftarrow \frac{n}{2^k}$$

$$\boxed{\frac{n}{2^k} = 1}$$

$$n = 2^k \leftarrow \frac{n}{2^k} + \frac{n}{2^k} \quad T(n) = k$$

Base condition

$$\Rightarrow k = \log_2 n$$

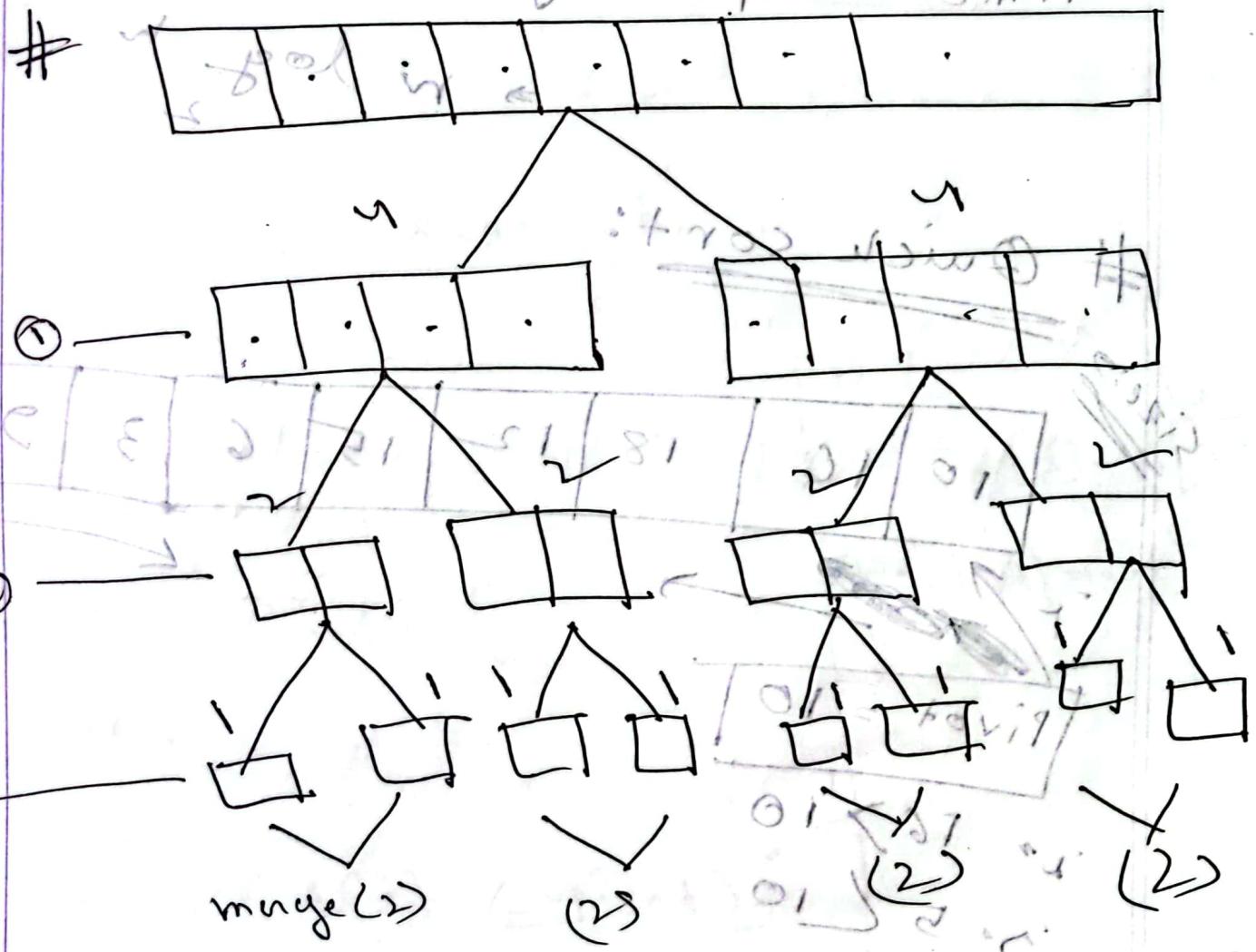
Ans

$$T(w) = 2^k + (1) + kw \quad \{ \leftarrow 8$$

$$\approx w + (\log_2 w)^w$$

$$T(w) \rightarrow wO(\log w \log \log_2 w) + N$$

recursion tree method:



$$\text{Merge} = \text{length}(sub1) + \text{length}(sub2)$$

$$8 \rightarrow 3 \quad w \sqrt{n} + 1 + \frac{w}{2} = (w) T$$

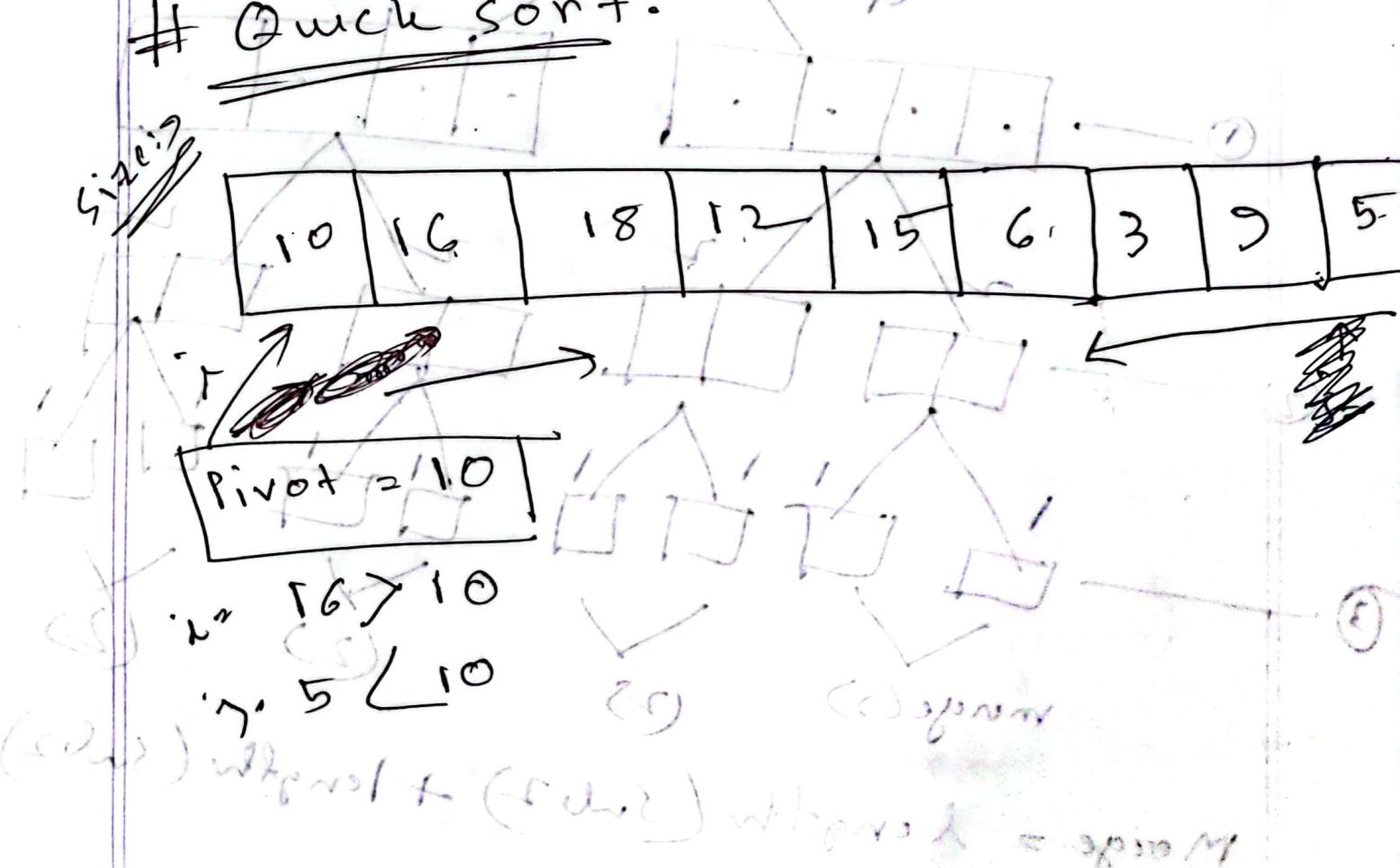
$$2^n \rightarrow n^w (n, b=1) + w$$

$$\Rightarrow k = (\log_2 n) \approx 30 \leftarrow (n) T$$

Time complexity $\rightarrow k \approx \log_2 n$

$$n \log_2 n$$

Quick sort:



$$\text{Pivot} = 10$$

$$16 > 10$$

$$5 < 10$$

(left) without (far) without swap

(right) without (far) without swap

(middle) without (far) without swap

(left) without (far) without swap

(right) without (far) without swap

(middle) without (far) without swap

(left) without (far) without swap

(right) without (far) without swap

(middle) without (far) without swap

20	5	18	9	15	6	3	9	16
----	---	----	---	----	---	---	---	----

L.W
13.6.23

PSA 2

Quick sort Algorithm

partition (l, h) {

 pivot = $A[l]$;

$i = l$; $j = h$;

 while ($i < j$) {

 do {

$i++$;

 while ($A[i] \leq \text{pivot}$);

 do { $j--$ }

 while ($A[j] > \text{pivot}$);

 if ($i < j$) {

 swap ($A[i], A[j]$);

 swap ($A[\text{pivot}], A[j]$);

 }

Quicksort (l, h) {

if (l, h) L [21 | 8 | 3 | 9]

j = partition (l, h);

Quicksort (l, j);

Quicksort ($j+1, h$);

~~Next case: 1 swap A from A[1]~~

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

$$\text{Worst case: } T(n) = O(n^2)$$

$$= (n-1) \times n$$

$$= n^2 - n$$

Algorithm → Quick Sort

→ Divide & Conquer

(Divide & Conquer)

G.W
G. 23

How does the descending order Quick sort algorithm work on the following data?

10	39	21	1	2	57	5
----	----	----	---	---	----	---

Pivot \rightarrow 10

$i > P$, $j < P$

10	5	21	1	2	57	39
----	---	----	---	---	----	----

This is for Ascending

10	5	2	2	21	57	39
----	---	---	---	----	----	----

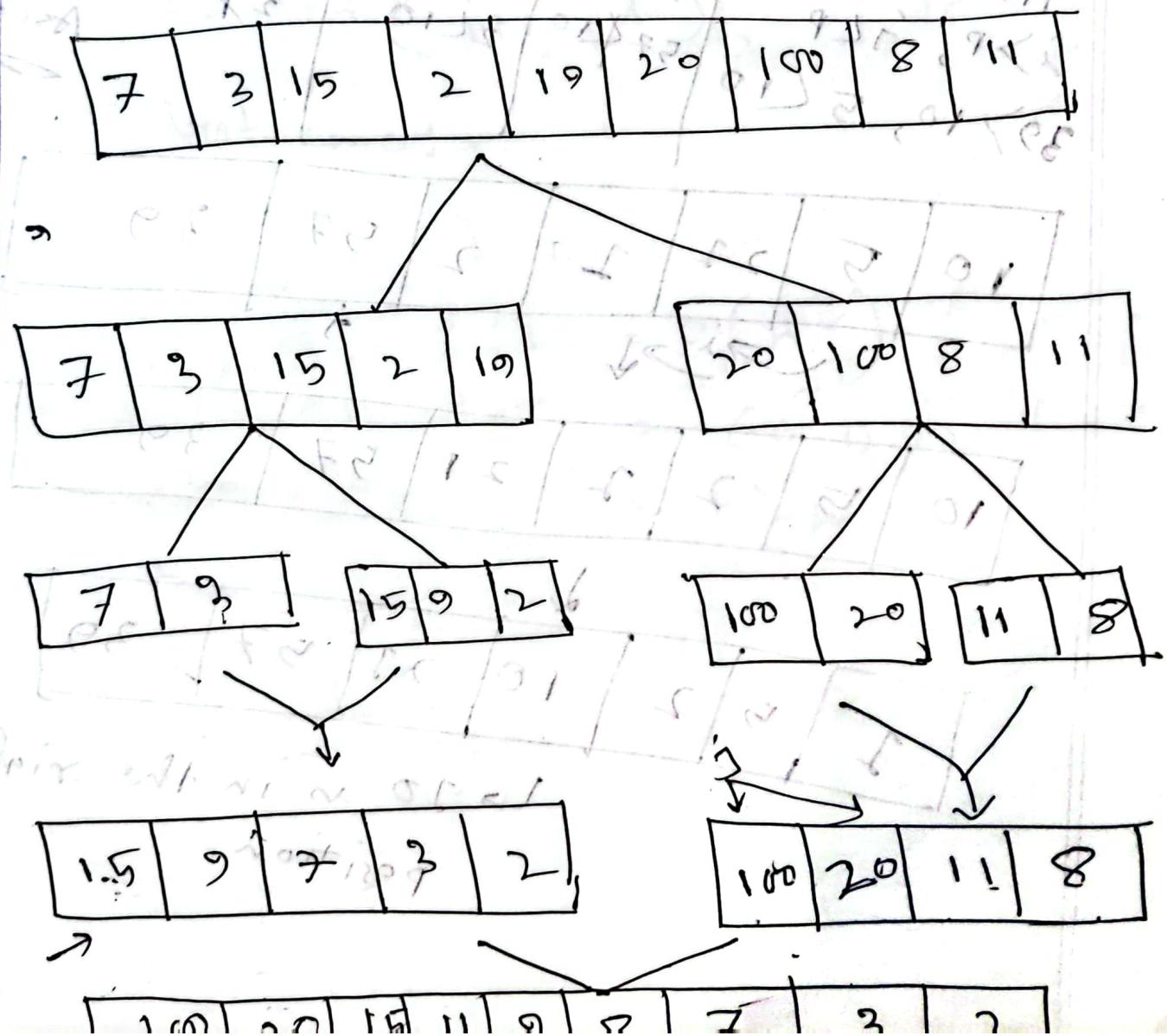
1	5	2	10	21	57	39
---	---	---	----	----	----	----

\hookrightarrow 10 is in the right position.

8	11	28	39
---	----	----	----

5	8	11	28
---	---	----	----

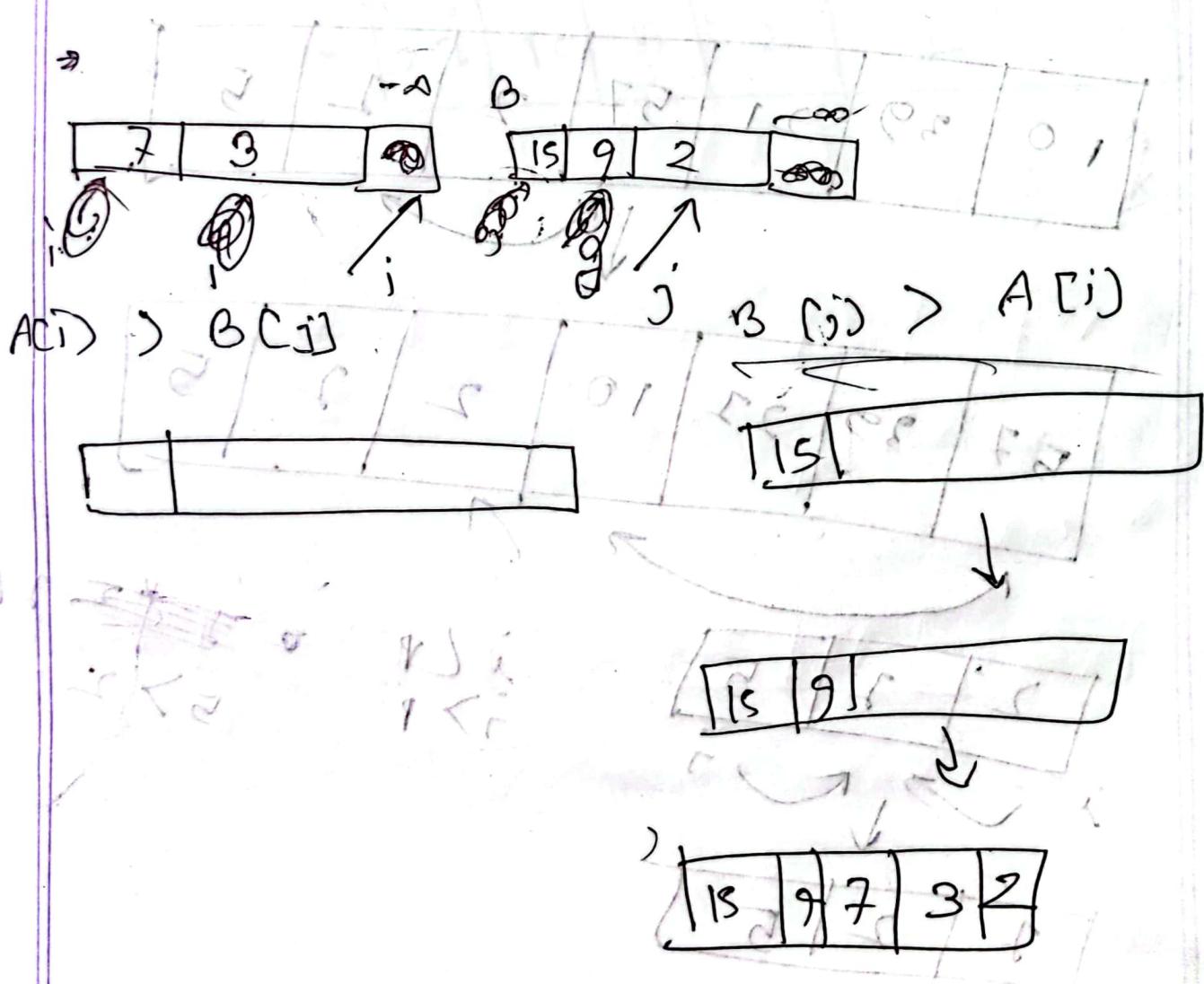
How does the merge sort algorithm work on the following data if you sort the array in descending order. Also show the merge step with proper i, j values for the last step.



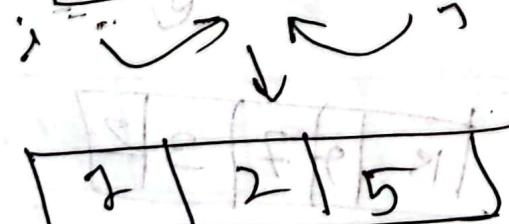
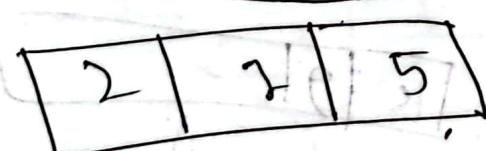
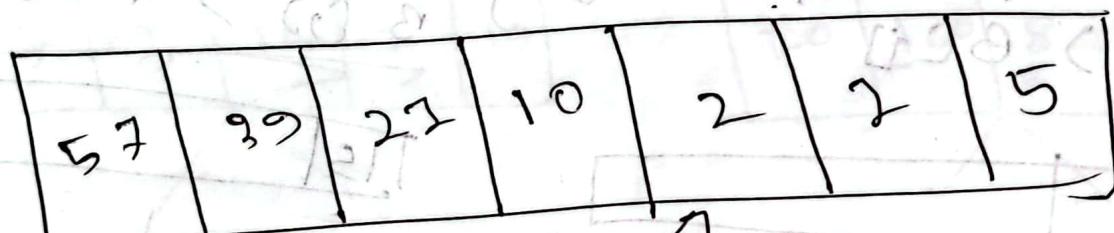
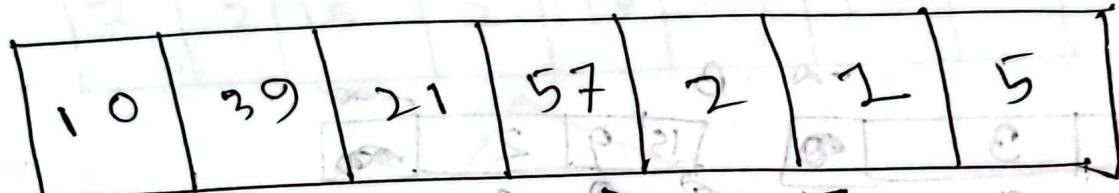
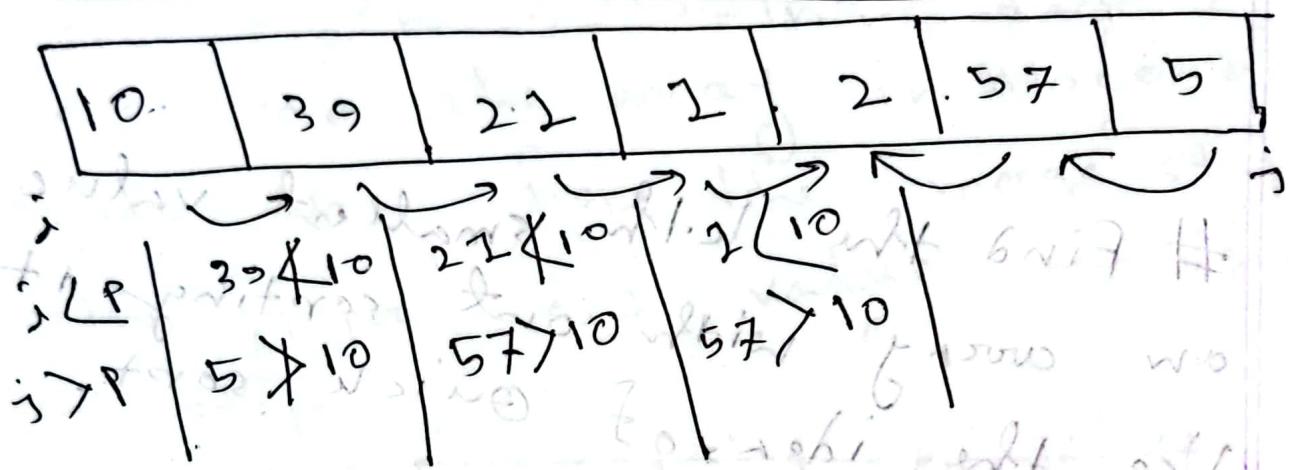
problem to solve

steps to solve it

Find the k-th smallest value in an array without sorting it.
Use the idea of quicksort.



Quick sort descending

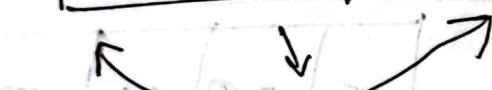


P.T.O.

2	2	5
---	---	---



5	2	2
---	---	---



57	39	22	10	5	2	2
----	----	----	----	---	---	---

$i < p$ ~~$i > p$~~ $2 < 2$
 $i > p$ ~~$i < p$~~ $5 > 2$

(c) $\Delta \text{S} \text{A}$ *

2	3	11	8	21
---	---	----	---	----

2	5	8	18	20
---	---	---	----	----

(c) $\Delta \text{S} \text{A}$ *

18	5	8	21
----	---	---	----

2 < 5
5 < 8
8 < 21

$A[i]$

15	9	7	3	2
----	---	---	---	---

i

$B[j]$

100	20	11	8
-----	----	----	---

j

* $A[i] < B[j]$

100	9	7	3	2
-----	---	---	---	---

i

15	20	11	8
----	----	----	---

j

* $A[i] < B[j]$

100	20	7	3	2
-----	----	---	---	---

i

15	9	11	8
----	---	----	---

j

* $A[i] < B[j]$

100	20	11	3	2
-----	----	----	---	---

i

15	9	7	8
----	---	---	---

j

* $A[i] < B[j]$

10	20	11	8	2	i
15	9	7	8	3	j

* $A[i] < B[j]$ with both

(working \rightarrow swap) i, j

(noting A, works)

(working \rightarrow swap) i, j swap

(noting A, it swaps with both)

(working \rightarrow swap) i, j swap

(noting A, it swaps with both)

(working \rightarrow swap) i, j swap

12

13

5	7	3	10	12	14	13
---	---	---	----	----	----	----

find nth element (A, l, h) {

if ($k-1 \geq position$) {

return $A[position]$;

}

else if ($k-1 < position$) {

find nth element ($A, l, position-1$);

}

else {

find nth element ($A, position+1, n$);

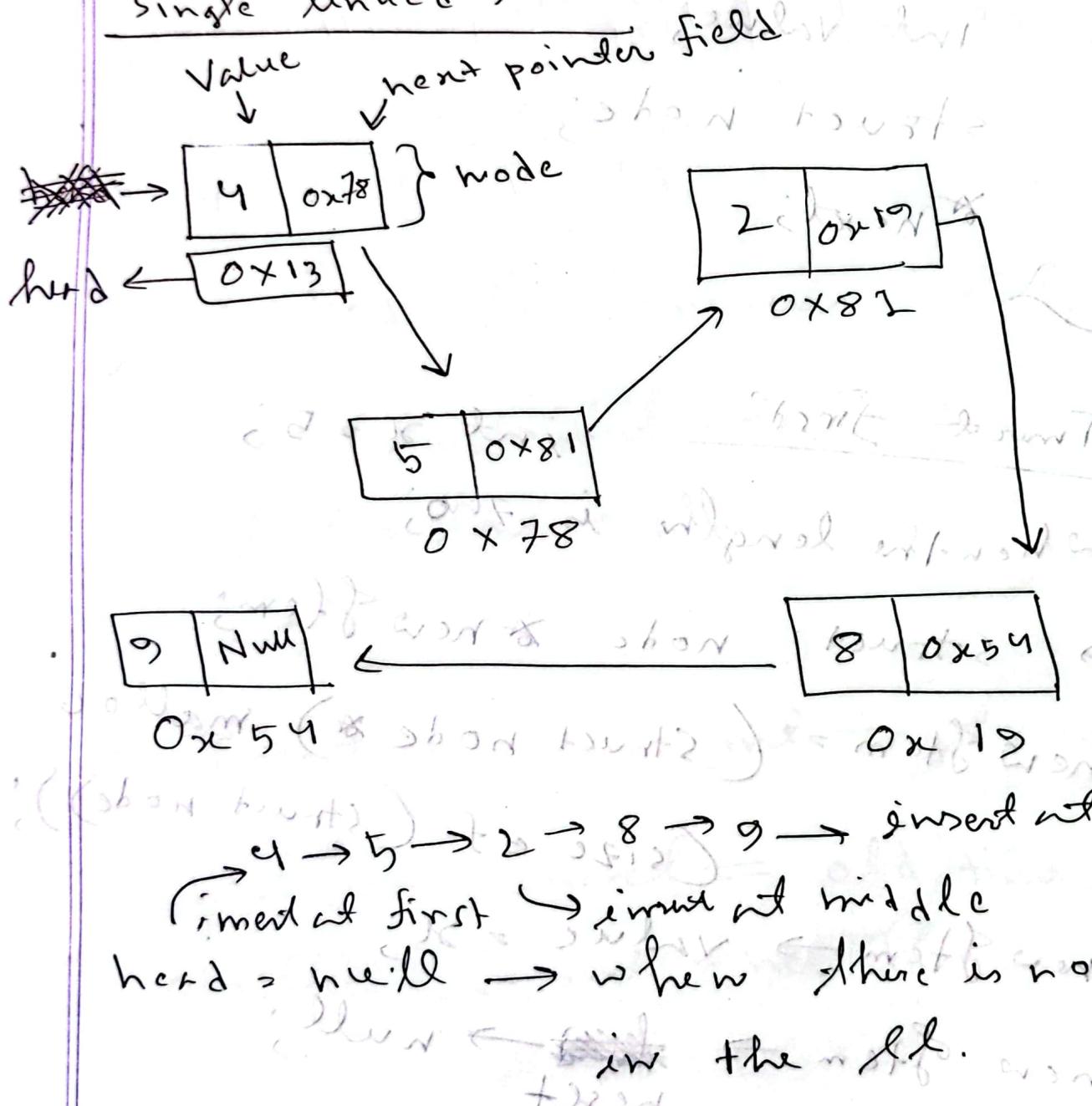
}

W
20.6.23

DSA

Linked list

Single linked list



```
struct node {  
    int value;  
    struct node* next;  
};
```



Insert first:

When the length is zero;

```
→ struct node *newItem;
```

```
newItem = (struct node*) malloc  
(sizeof(struct node));
```

```
newItem → value = x;
```

```
newItem → next → null;  
next
```

259

when the length is not zero: →

new item → head; ↗
head = new item;

head = head + 1; ↗
head = head + 1; ↗

linked list traverse basic idea:-

while (lmt → next != Null) { ↗

lmt = lmt → next; ↗
→ ↗
insert lmt;

Insert At middle (int target, int x) { ↗

new item → next = old item → next

old item → next = new item;

P.T.O.

$7 \rightarrow 4 \rightarrow 5 \rightarrow 2 \rightarrow 8 \rightarrow 9 \rightarrow 10$

```
search (int y){  
    target = head;  
    while (target != null){  
        if (target->value == y){  
            flag = 2;  
        }  
    }  
    else {  
        target = target->next;  
    }  
    if (flag == 2){  
        flag++;  
    }  
}
```

Delete at first:

$3 \rightarrow 5 \rightarrow 2 \rightarrow 9$

↑

head
current = head;

head = head → next;

free (current);

6/1
24.6.23

~~Delete last:~~

$2 \rightarrow 7 \rightarrow 8 \rightarrow 2 \rightarrow 2$

current
Previous

* current = head;

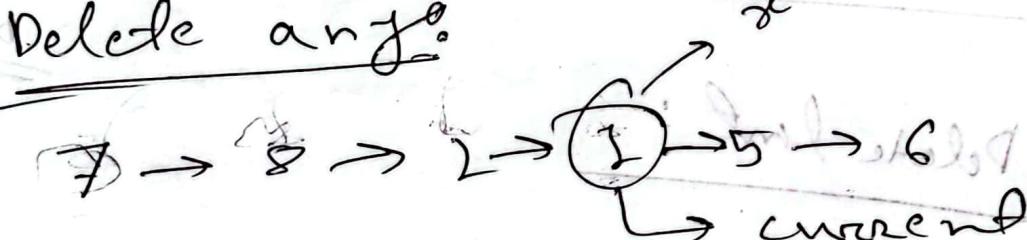
* prev = null;

while (current → next != null) {
 prev = current;
 current = current → next;

prev → next = null;
 free (current);

~~if~~ if (~~prev != null~~) {
 head = null; ~~s = s + s~~
 else {
~~(head = head.next)~~
 prev = ~~next = next.next~~ = head
 }
 }
 } (Changes) ~~not~~

Delete any:



~~prev~~ → ~~current~~ → ~~next~~
~~prev~~ → ~~current~~ → ~~next~~
~~prev~~ → ~~current~~ → ~~next~~

while (~~current != null~~ & ~~x~~) {

~~prev = current~~;

~~current = current.next~~;

~~if x == current.data~~

~~then x = null~~

```
if (prev != null) {  
    prev->next = current->next;  
    else { head = null;  
        free (current);  
    }  
}
```

Double linked list

```
struct dnode{
```

```
    int value;
```

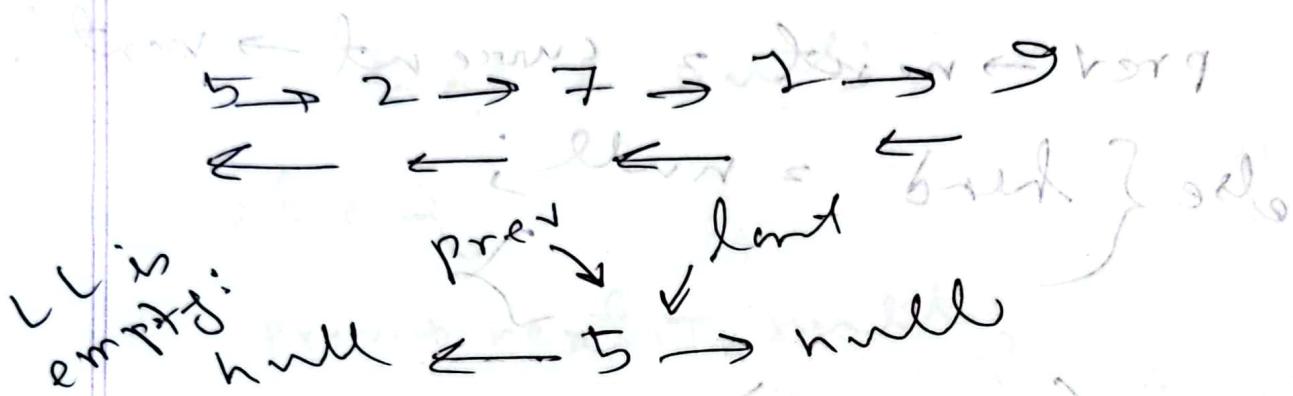
```
    struct dnode *next;
```

```
    struct dnode *prev;
```

```
}
```

P7.0

Insert first (LL & LWT) 27



head = new item;

head → next = null;

head → previous = null;

last = head;

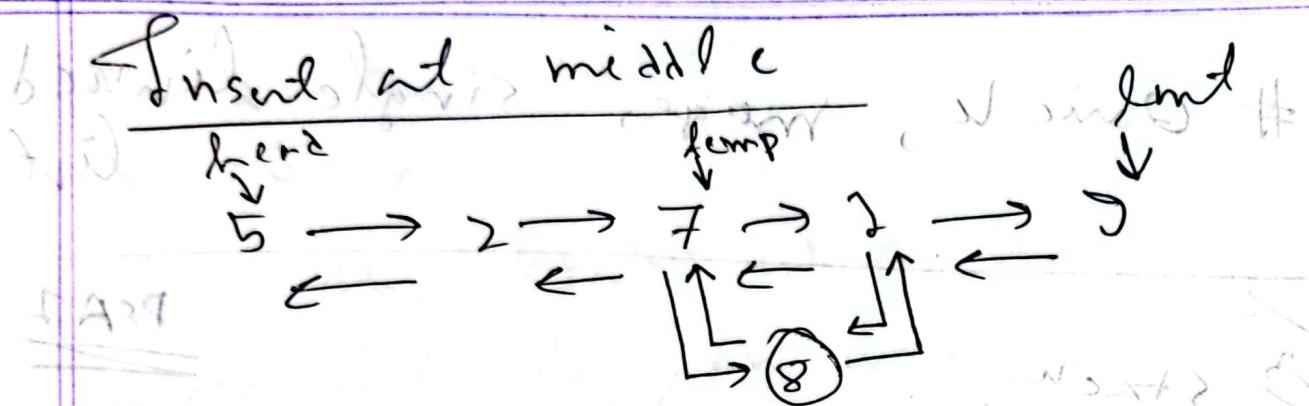
LL is not empty:

new item → previous = null;

head → prev = new item;

new item → next = head

head = new item;



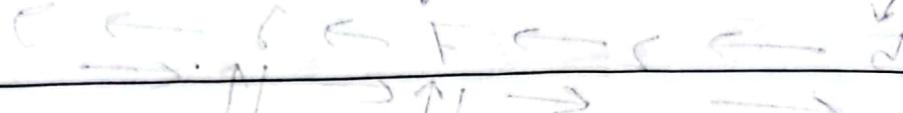
temp = head; // initial value
 while (temp != null && temp.next.value < 8) {
 temp = temp.next;
 }

temp = temp.next; // after while loop
 new item → next = temp.next; // new item's next is temp's next
 new item → prev = temp; // new item's previous is temp

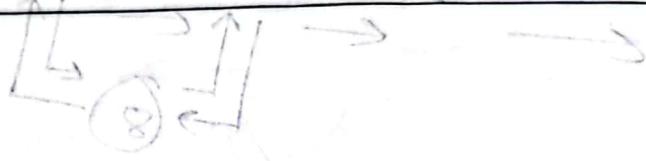
temp.next → prev = new item; // temp's next is new item

temp → next = new item; // temp's next is new item

Quick, merge, single linked list



~~GW
8.7.23~~ stack



PsA1

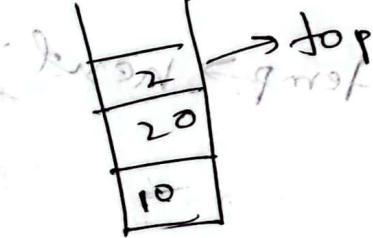
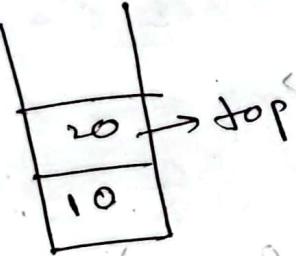
* stack is LIFO data structure

Last in first out

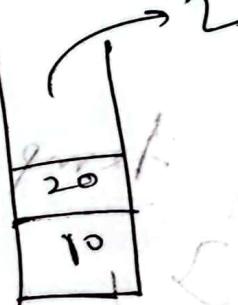


top

push (10)



pop()



2

push (20)
push (2)

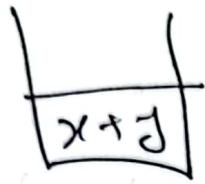
last element from stack

$$x = 2$$
$$j = 7$$

Show the status of a stack
implement by an array [7] for
the operations given below:

- | | |
|---|--|
| <ul style="list-style-type: none">① push($x+j$)② push(x)③ pop()④ push($j+x$)⑤ pop()⑥ pop()⑦ push($8+x$) | <ul style="list-style-type: none">⑧ push($9+x$)⑨ push($j+x$)⑩ push($8+x$)⑪ push($10+x$)⑫ push(20)⑬ (pop) |
|---|--|

①



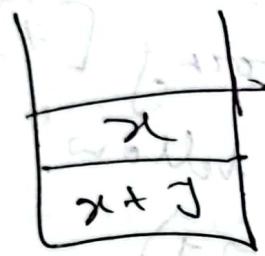
top = 0

base = 26 8

F = 6

Value to be entered into stack

②

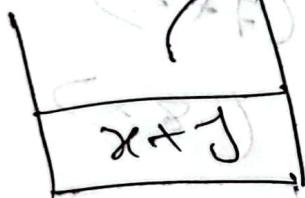


top = 2

Time signature

Value writings left

③

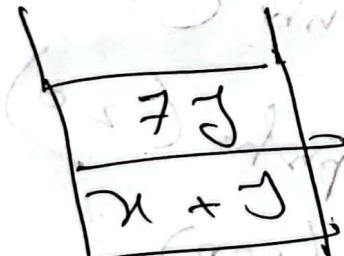


top = 0

(base) day ①

(x) day ②

④

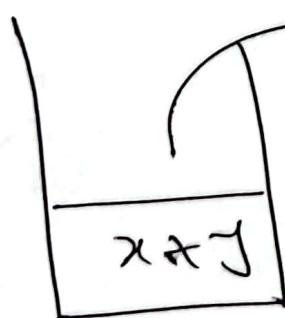


top = 2

day ③

day ④

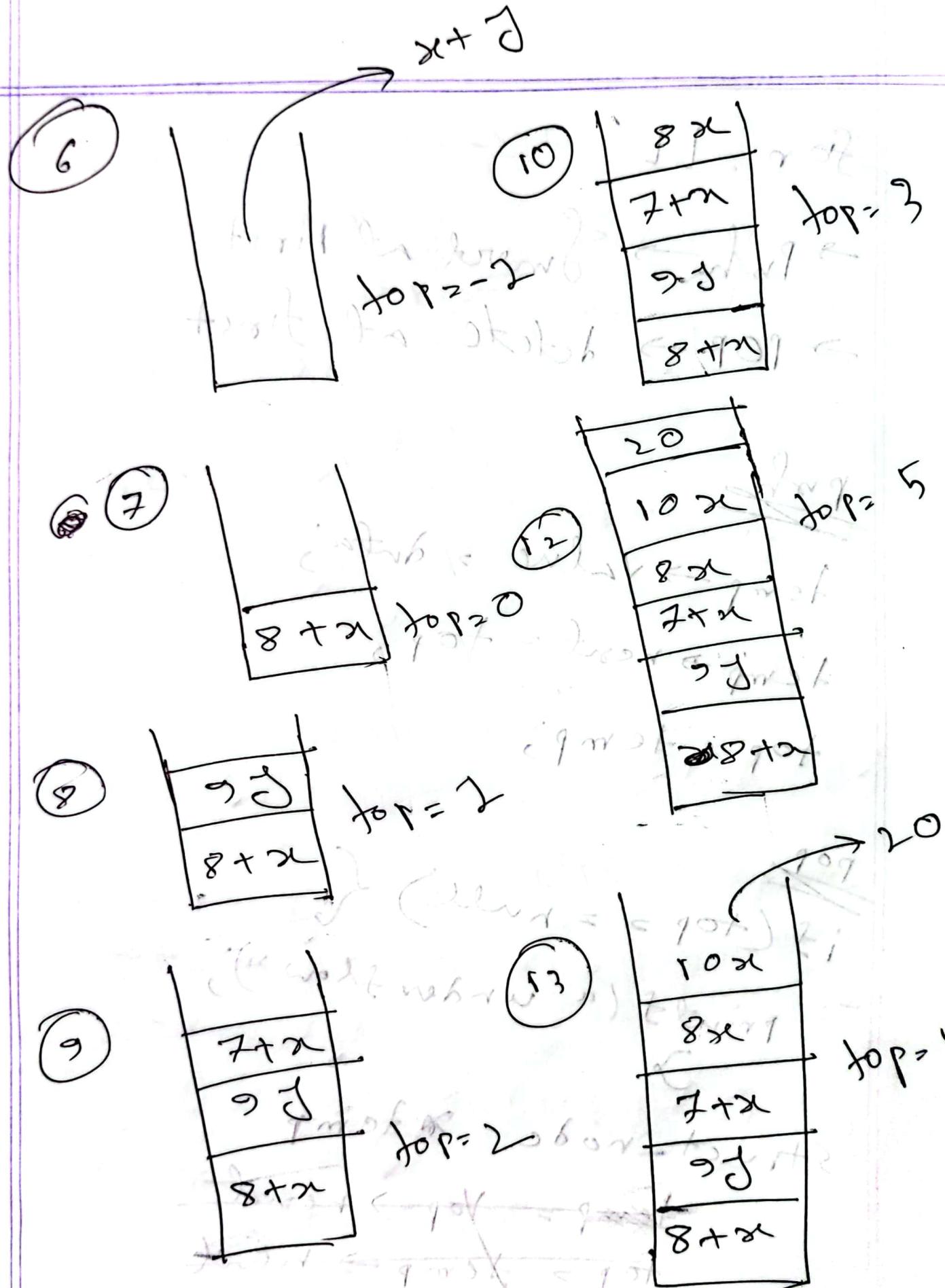
⑤



7j

day ③

day ④



for LL

→ Push → Insert at First

→ POP → Delete at first

~~push~~

temp → value → data;

temp → next = top;

top = temp;

~~POP~~

if (top == null) {

printf("Underflow");

}

struct nodes *temp

~~temp = top → next~~

~~top = temp → next~~

~~data~~ = top → value (cond) step ①

temp = top

top = top → next (s) step ②

free (temp);

return data;

2. ~~data~~ = top → value (t) step ③

~~data~~ = top ← (s) step ④

top = ~~data~~ ← (s+8) step ⑤

top = ~~data~~ ← (s+8) step ⑥

top = ~~data~~ ← (s+8) step ⑦

top = ~~data~~ ← (s+8) step ⑧

top = ~~data~~ ← (s+8) step ⑨

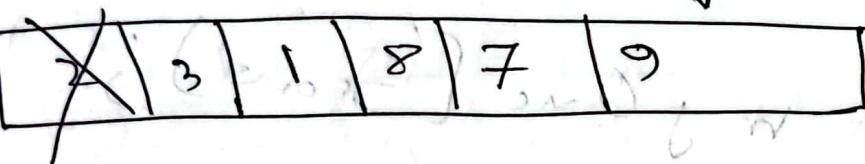
top = ~~data~~ ← (s+8) step ⑩

- ① $\text{push}(x+y) \rightarrow$ ~~$x+y$~~ $\leftarrow \text{top} = x+y$
 top
- ② $\text{push}(x) \rightarrow$ ~~x~~ $\leftarrow \text{top} = x$
 top
- ③ $\text{pop}() \rightarrow$ ~~$x+y$~~ $\leftarrow \text{top}$
 top
- ④ $\text{push}(7j) \rightarrow$ ~~$7j$~~ $\rightarrow x+y$
 top
- ⑤ $\text{pop}() \rightarrow$ $x+y$
 top
- ⑥ $\text{pop}() \rightarrow$ null^i
 top
- ⑦ $\text{push}(8+x) \rightarrow$ $8+x$
 top
- ⑧ $\text{push}(9j) \rightarrow$ $9j \rightarrow 8+x$
 top
- ⑨ $\text{push}(7+x) \rightarrow$ $7+x \rightarrow 9j \rightarrow 8+x$
 top
- ⑩ $\text{push}(8x) \rightarrow$ $8x \rightarrow 7+x \rightarrow 9j \rightarrow 8+x$

Queue → (FIFO)

first In First Out

front



enqueue (insert) → rear ++

dequeue (delete) → front ++

~~void~~ (insert)

void enqueue (int x) {

i. check overflow

ii. check if x is the first element

or not

iii. rear ++;

 myQue [rear] = x;

}

ii:



Delete

(0777) ← unused
int degree()

check underflow for reasons

~~if myOne[front] is~~

front + 1 is empty
} front = should be empty

* Sample Ques: →

P = 23, {Q = 7-P; } (Ans)

array size > 5. show the steps for the following operations on the Queue.

- ① Dequeue ()
- ② Enqueue (3P)
- ③ Enqueue (2e-P)

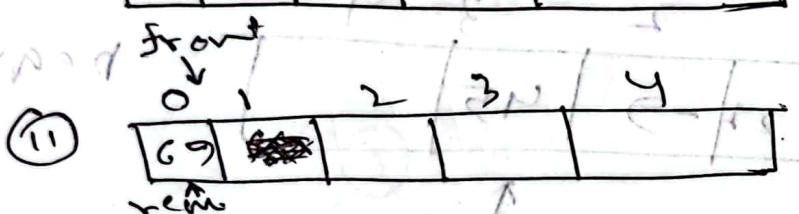
- IV Enqueue (7+P)
- V Dequeue ()
- VI Dequeue ()
- VII Enqueue (22+P)

⑧ Dequeue (\rightarrow) $\leftarrow 8 \rightarrow 5 \rightarrow 1 \rightarrow 0$ iv

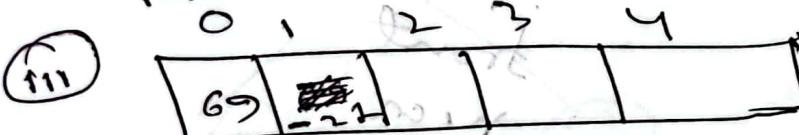
⑨ Dequeue (\rightarrow)



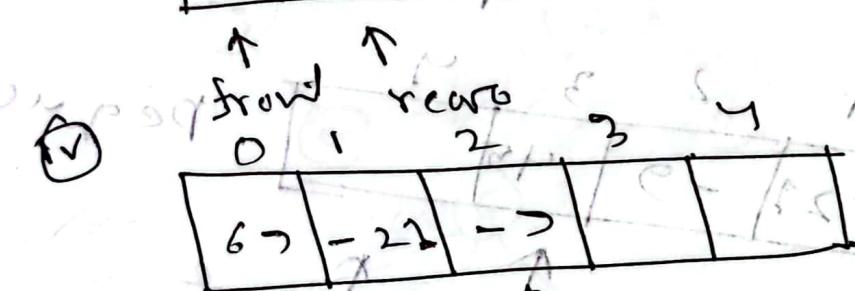
(unbalanced)



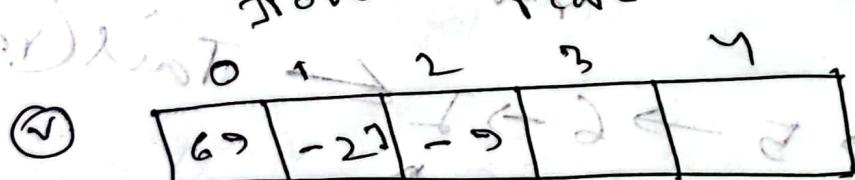
Enqueue (3 \rightarrow)



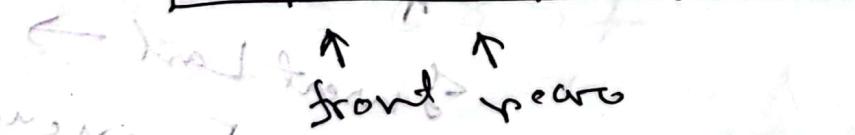
Enqueue (2 \rightarrow)



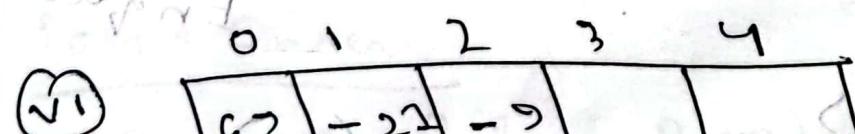
Enqueue (7 \rightarrow) v



Dequeue ()



Dequeue ()



Dequeue ()

front
rear

vii

0	1	2	3	4
69	-21	-7	45	

Enqueue()

22+P

↑ ↑

front rear

0	1	2	3	4

viii

0	1	2	3	4
69	-21	-7	45	

Dequeue()

ix

0	1	2	3	4
69	-21	-7	45	

Dequeue

head(front)

2 → 3 → 5 → 6 → 7

Tail(rear)

↑

insert last →

Delete first

→ Dequeue()

{O(n)}

H	E	S	S	,
e	-	s	s	O(n)

work
with

Enqueue()

iv

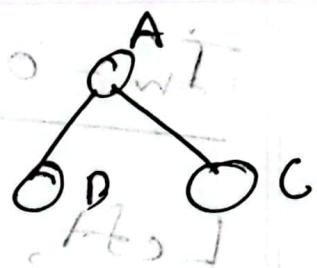
C.W
24.7.23

Trees & Tree traversals

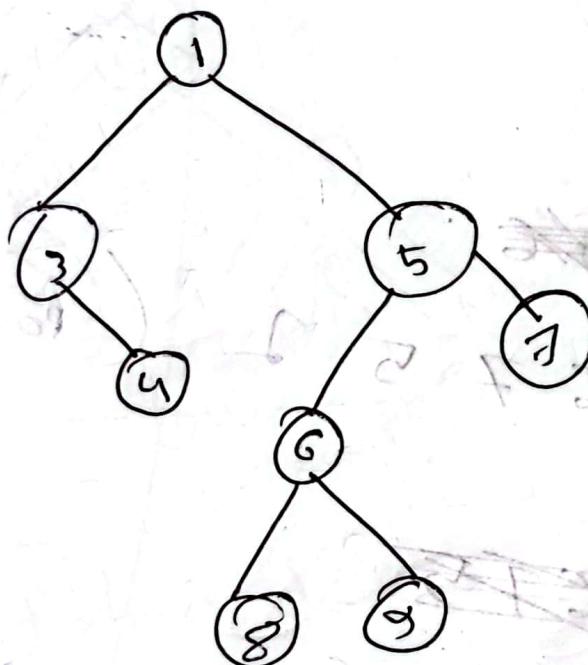
DSA 2

Pre-Order

Root, left, right



A B C



2 3 4 5 6 8 9 7

Post-Order

Left, Right, Root

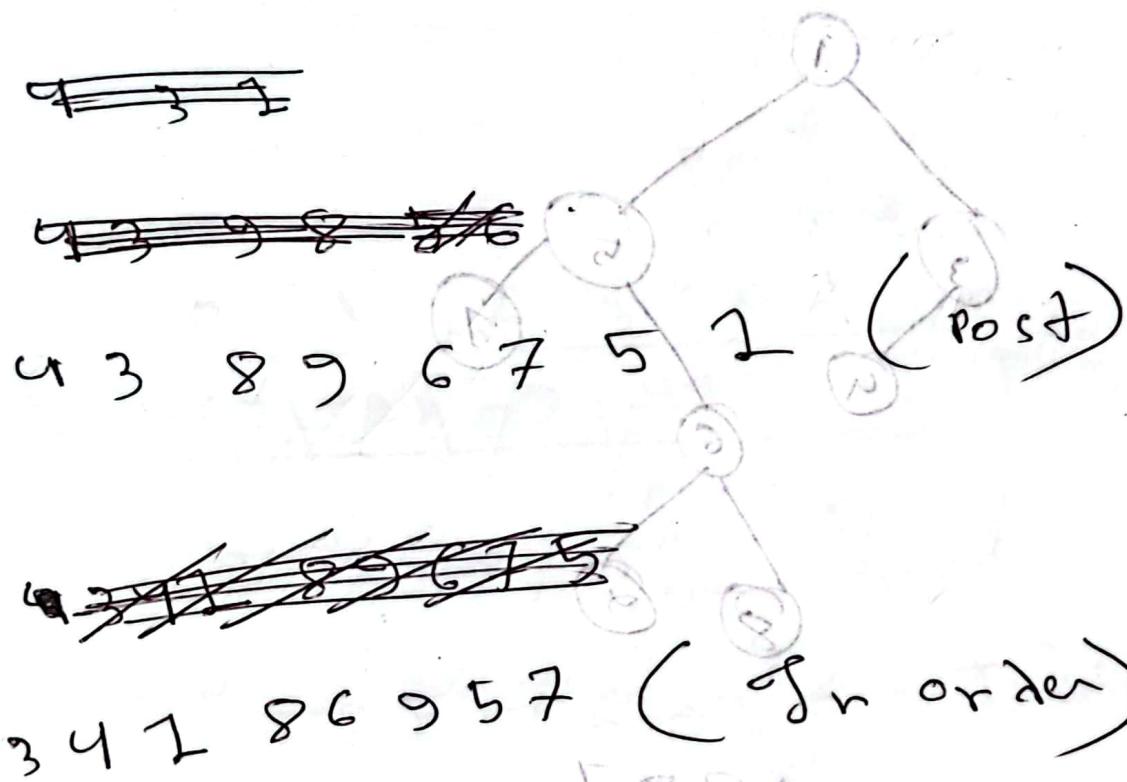
P.T.S

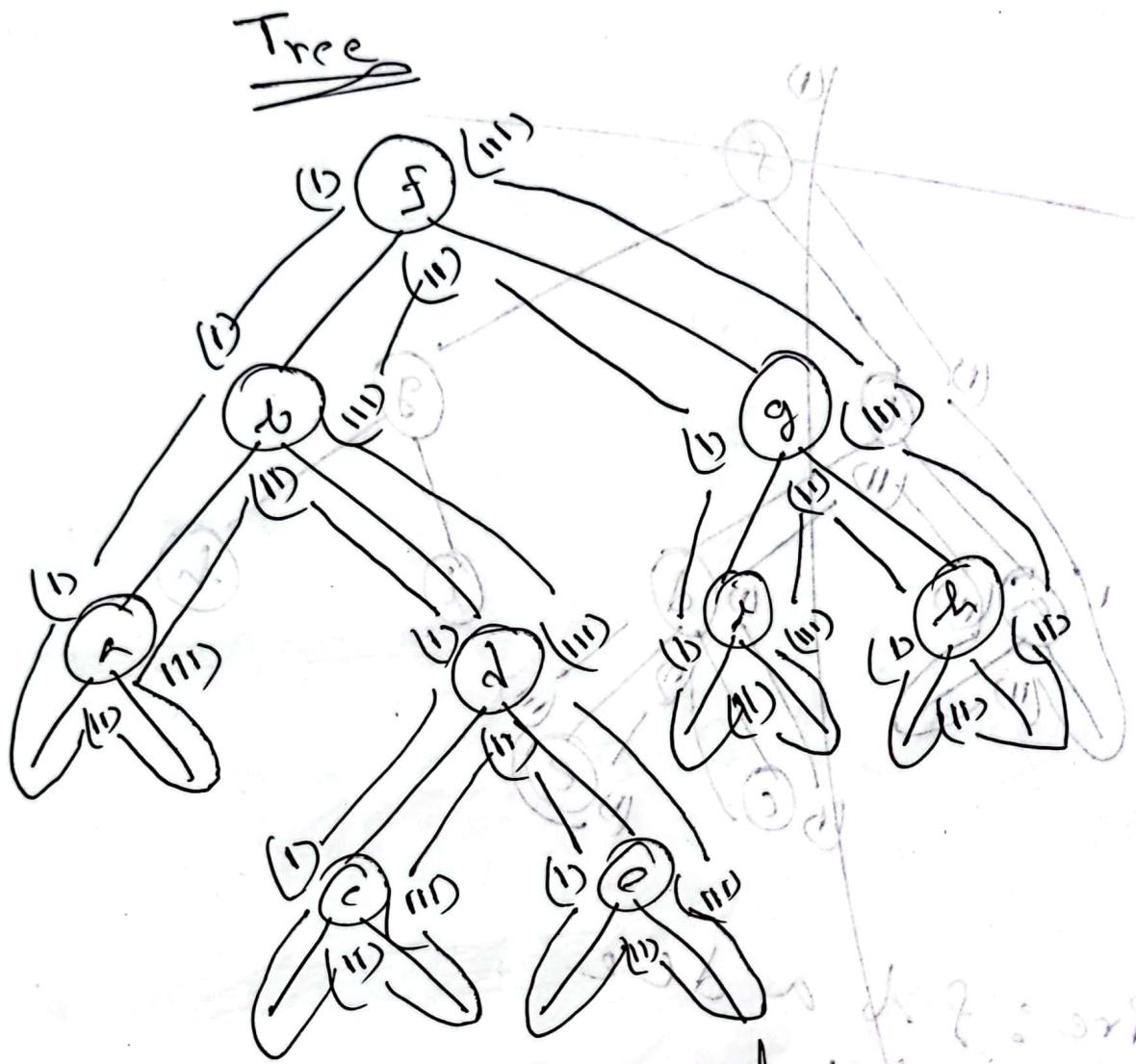
A)

Wanted sort do sort

In-order

Left, Root, Right





Pre: f b a d c e g i h
 In: a b c d e f i g h
 Post: a c e d b i h g f

in

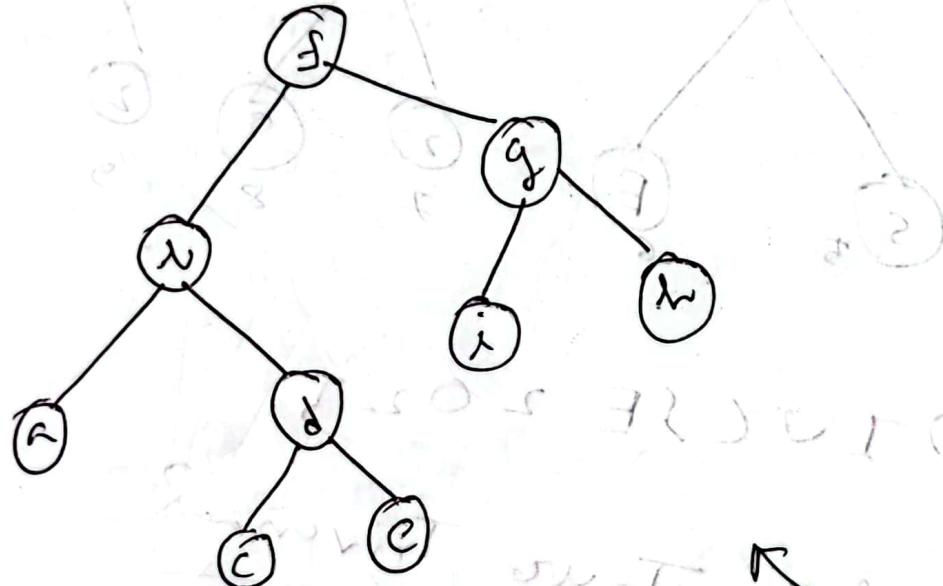
6/1

~~Tree~~

~~DSA 1~~

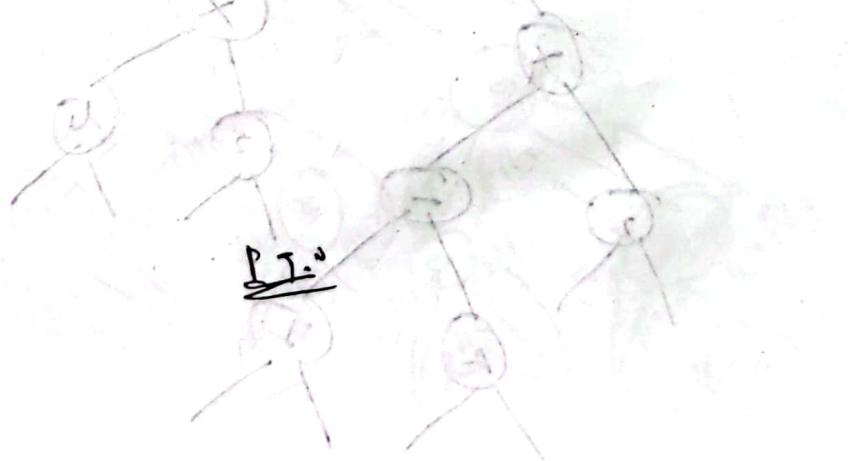
PreOrder: f → b → a → d → c → c → g → i → h

In: a → b → c → d → e → f → i → g → h



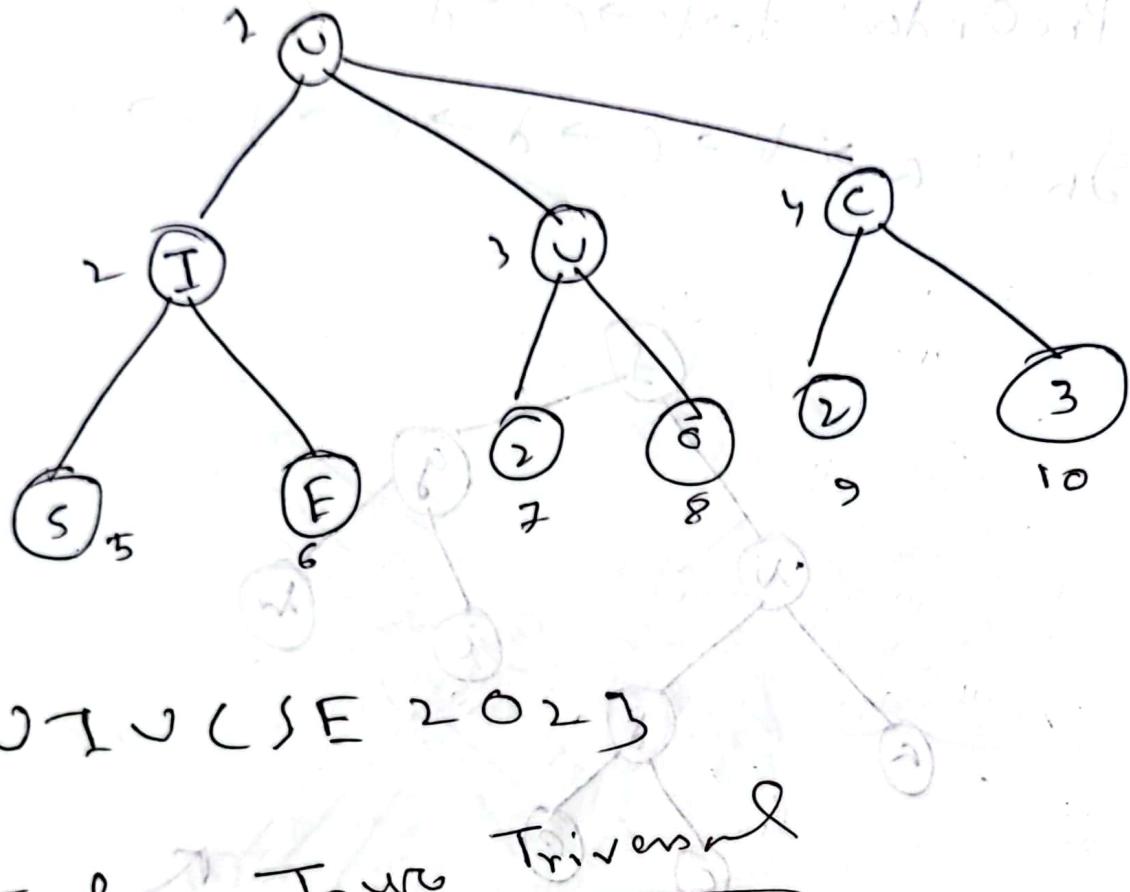
Post: a c e d b i h g f

in: a b c d e f i g h



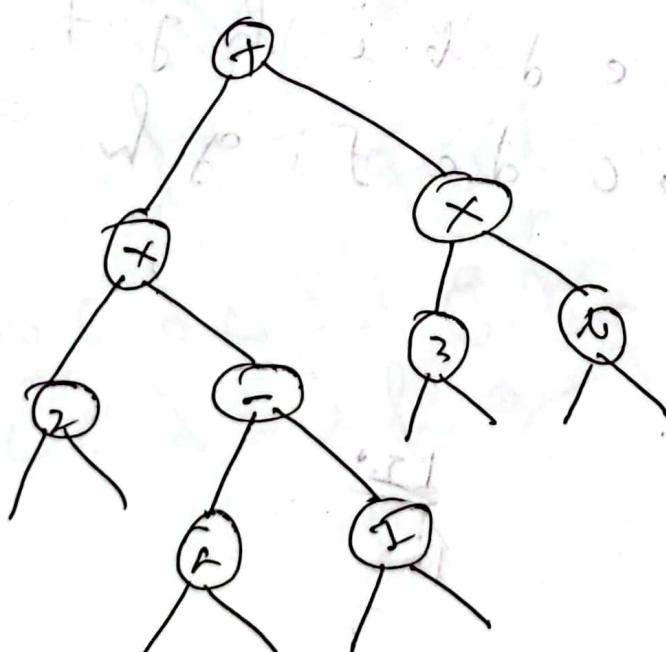
Level Order Traversal

*



→ VTU CSE 2023

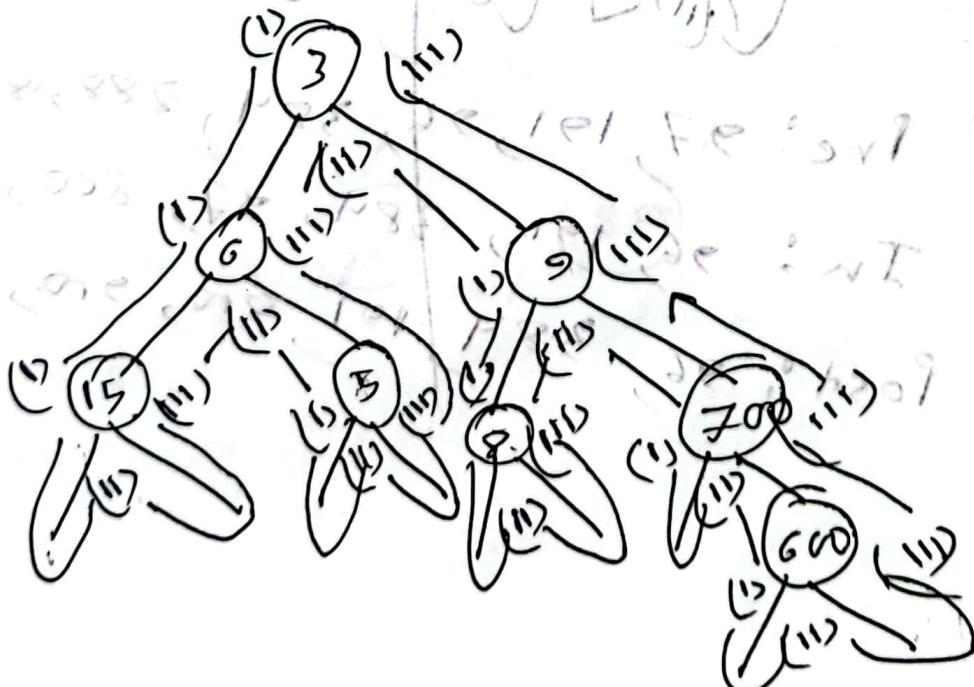
* Endian Tree Traversal



$$\text{In} : \left\{ 2 * (a - 1) + (3 + b) \right\}$$

* Draw A binary search tree;

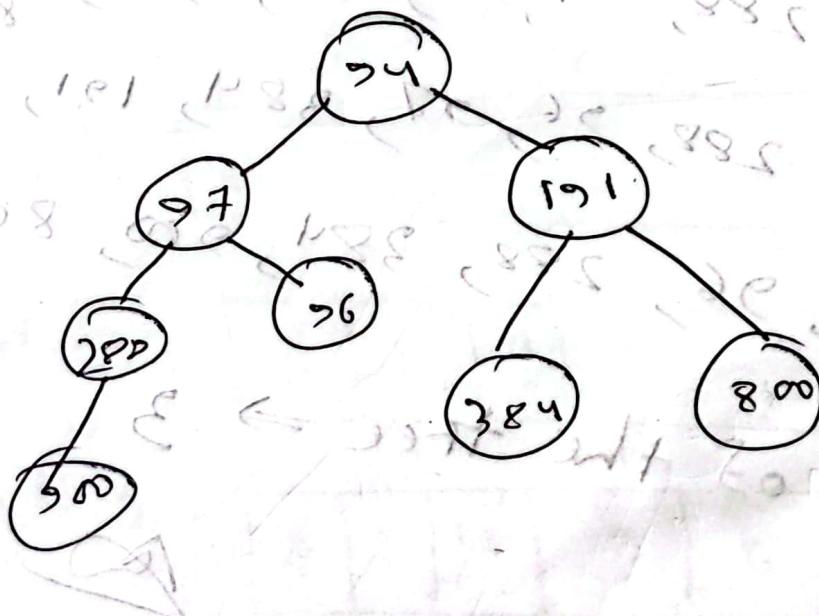
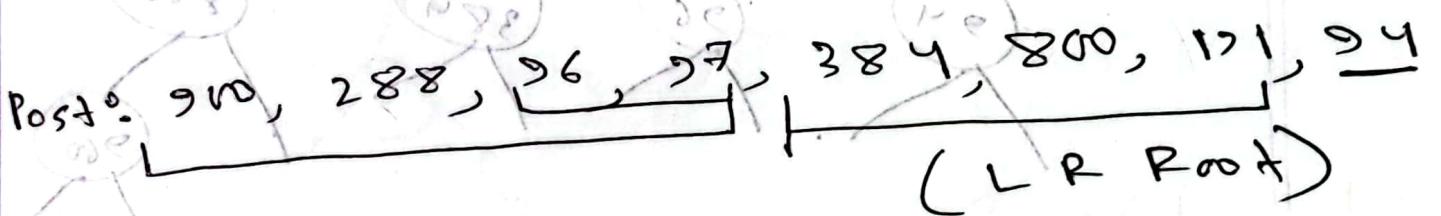
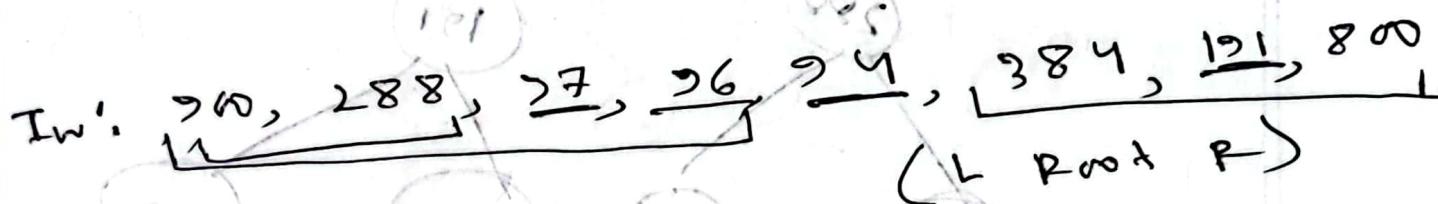
$$\begin{aligned}
 & x = 3, y = 6, z = 9, s = 5, t = 8, p = 15 \\
 & u = 600, v = 700 \\
 & \text{Pre} = 3, 6, 15, 5, 9, 8, 700, 600 \quad (\text{Root}^+, L, R) \\
 & \text{In} = 15, 6, 5, 3, 8, 700, 9, 600 \quad (L, \text{Root}^+, R)
 \end{aligned}$$

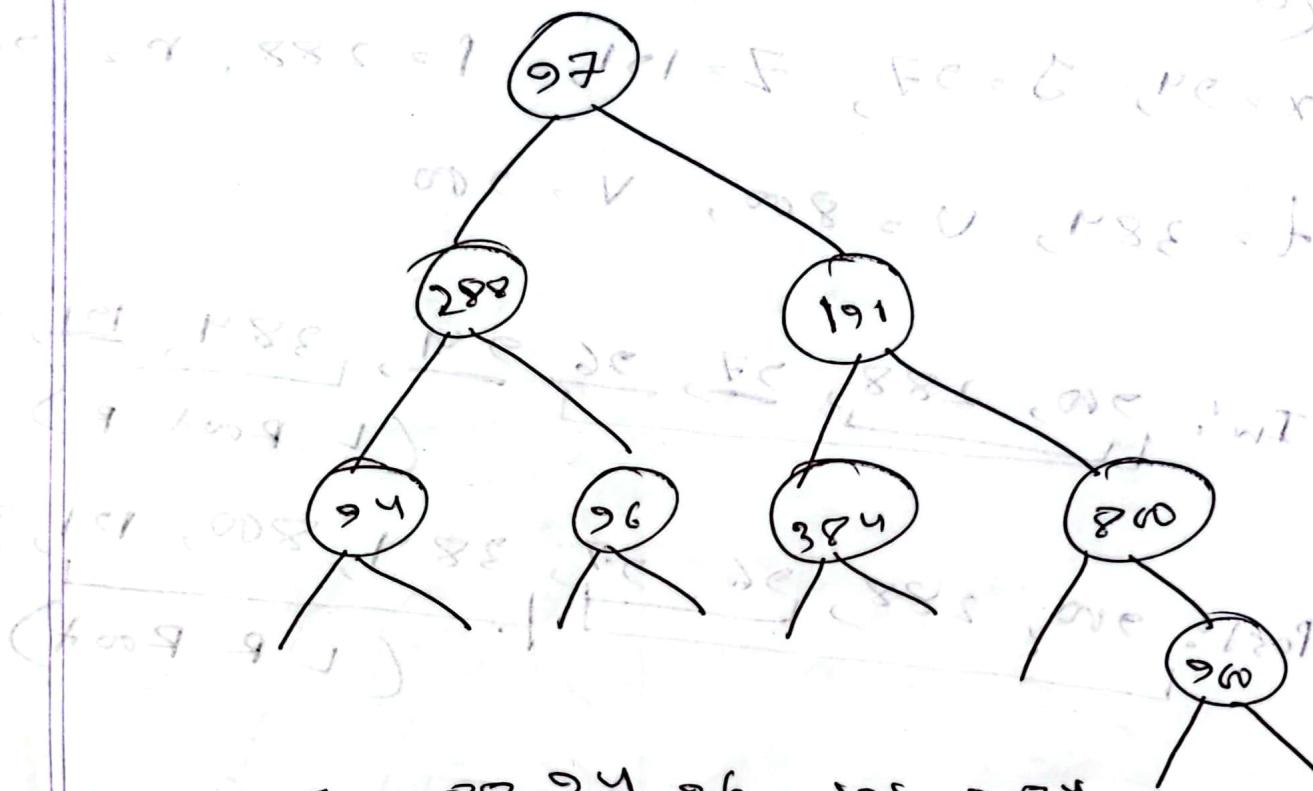


(6)

$$x = 74, \beta = 77, \gamma = 171, p = 288, r = 76,$$

$$t = 384, v = 800, w = 900$$





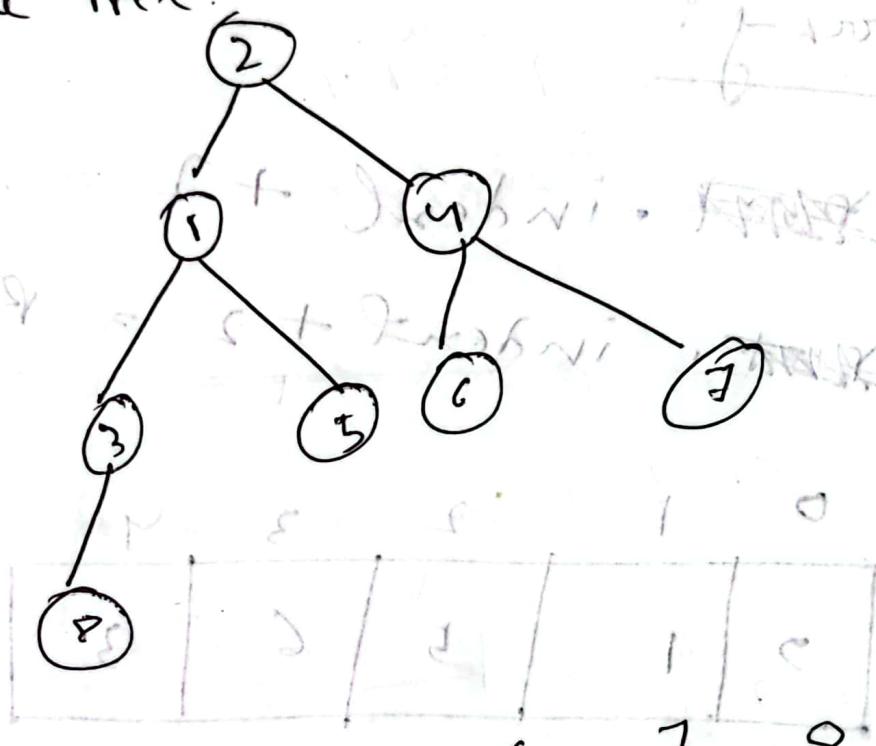
Pre: 97, 288, 94, 96, 191, 384, 800, 900,

In: 94, 288, 96, 97, 384, 191, 800, 900,

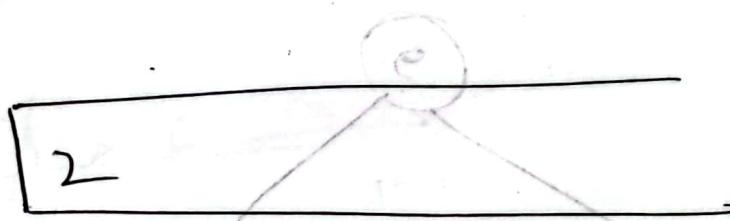
Post: 94, 96, 288, 384, 900, 800, 191, 97

height of the tree $\Rightarrow 3$

Q) Show the simulation of level order technique using a Queue for the tree:



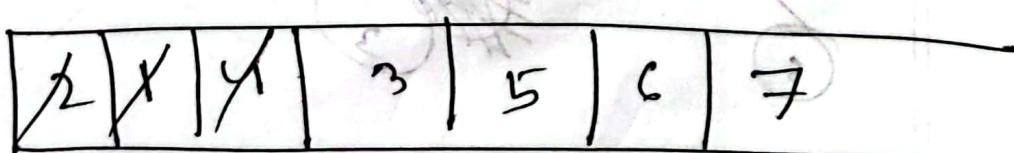
Level 0



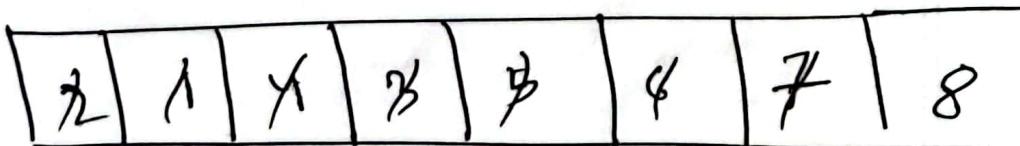
Level 1



Level 2



Level 3



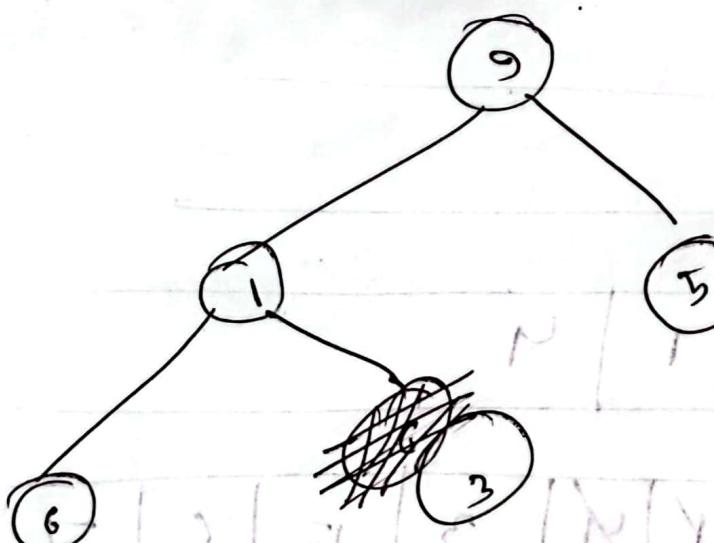
deg();
print();

array:

$$2 \cdot \text{index} + 1 = \text{left}$$

$$2 \cdot \text{index} + 2 = \text{right}$$

0	1	2	3	4
2	1	5	6	3



$$2 \cdot 0 + 1 = 1 \rightarrow L$$

$$2 \cdot 0 + 2 = 2 \rightarrow R$$

8	1	8	1	8	1	8	1	7	5
---	---	---	---	---	---	---	---	---	---

80 prob
82 train

~~C. 2
5. 8. 23~~

~~DSA - 2~~

in Case 2: $\{ \text{index} = 0 \}$

$P \rightarrow C$

$| 2i + 1 \Rightarrow L$

$| 2i + 2 \Rightarrow R$

$C \rightarrow P$

$\left\lfloor \frac{i-2}{2} \right\rfloor$

Case 2: $\{ \text{index} \rightarrow 2 \}$

$P \rightarrow C$

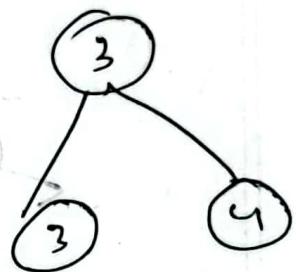
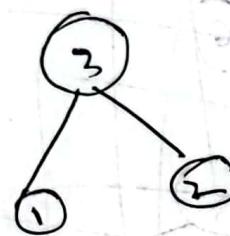
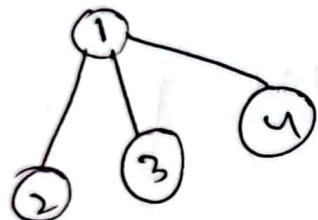
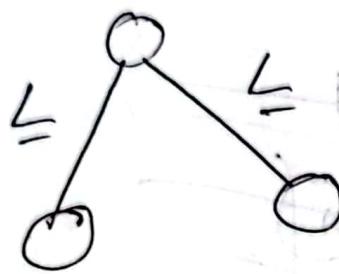
$2i$

$2i + i$

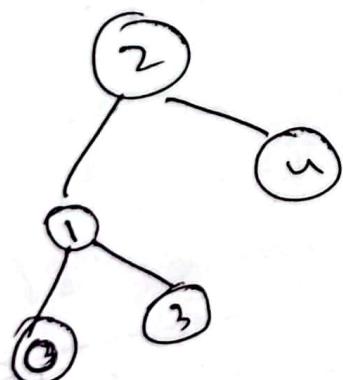
$C \rightarrow P$

$\left\lfloor \frac{i}{2} \right\rfloor$

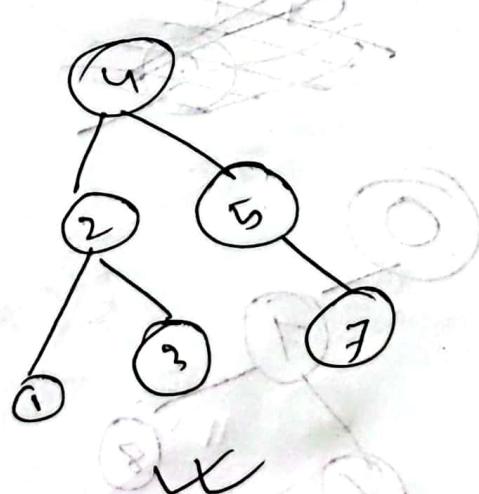
BST



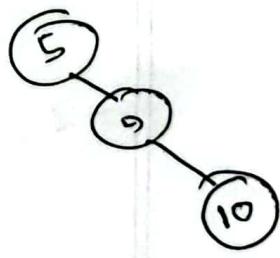
X



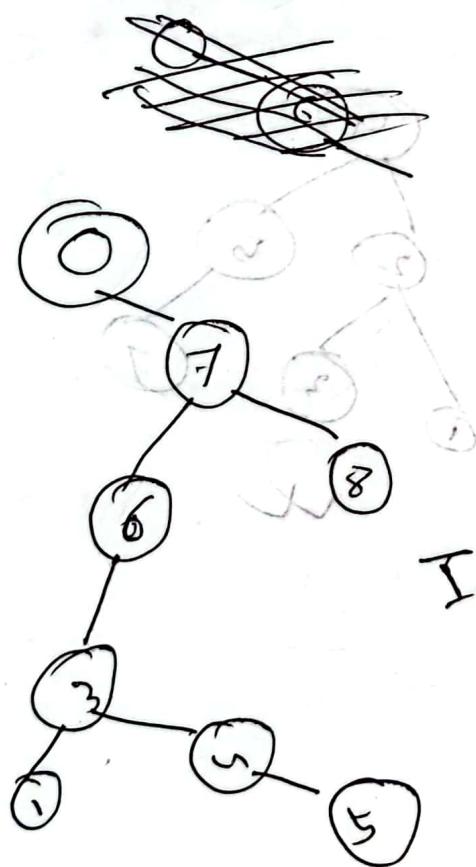
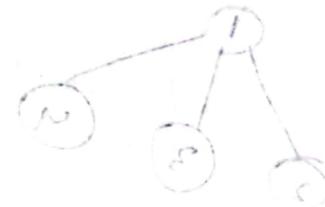
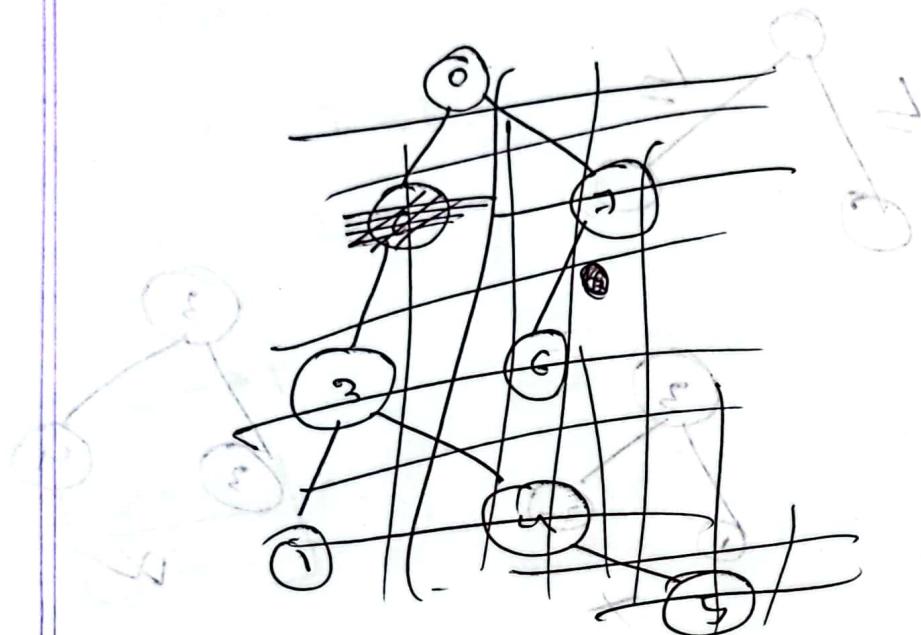
X



X

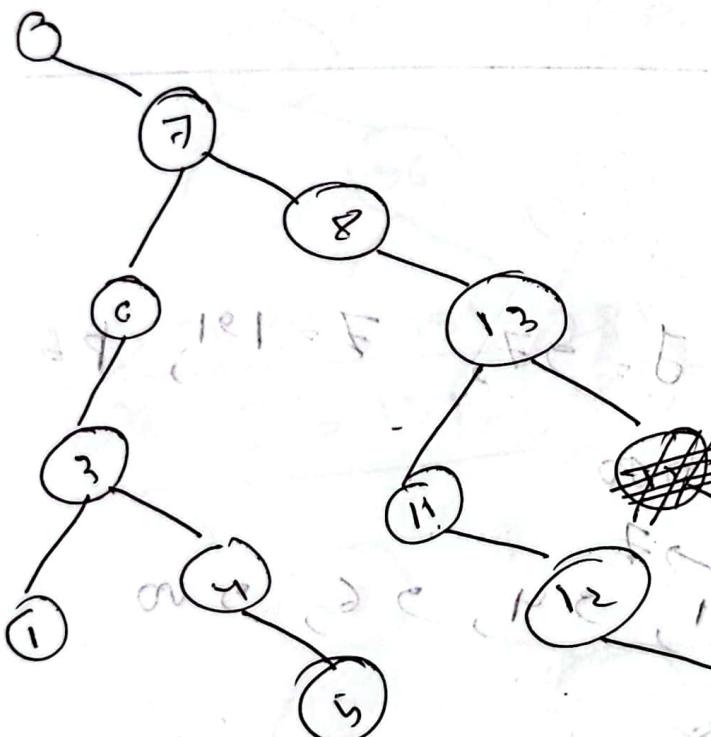


0, 3, 6, 3, 2, 4, 5, 8



In: 0, 1, 3, 4, 5, 6, 7, 8

0, 7, 6, 3, 1, 4, 5, 8, 13, 2, 14, 12,
 some are P.P
 to make tree



12 augment (+)

→ Tree

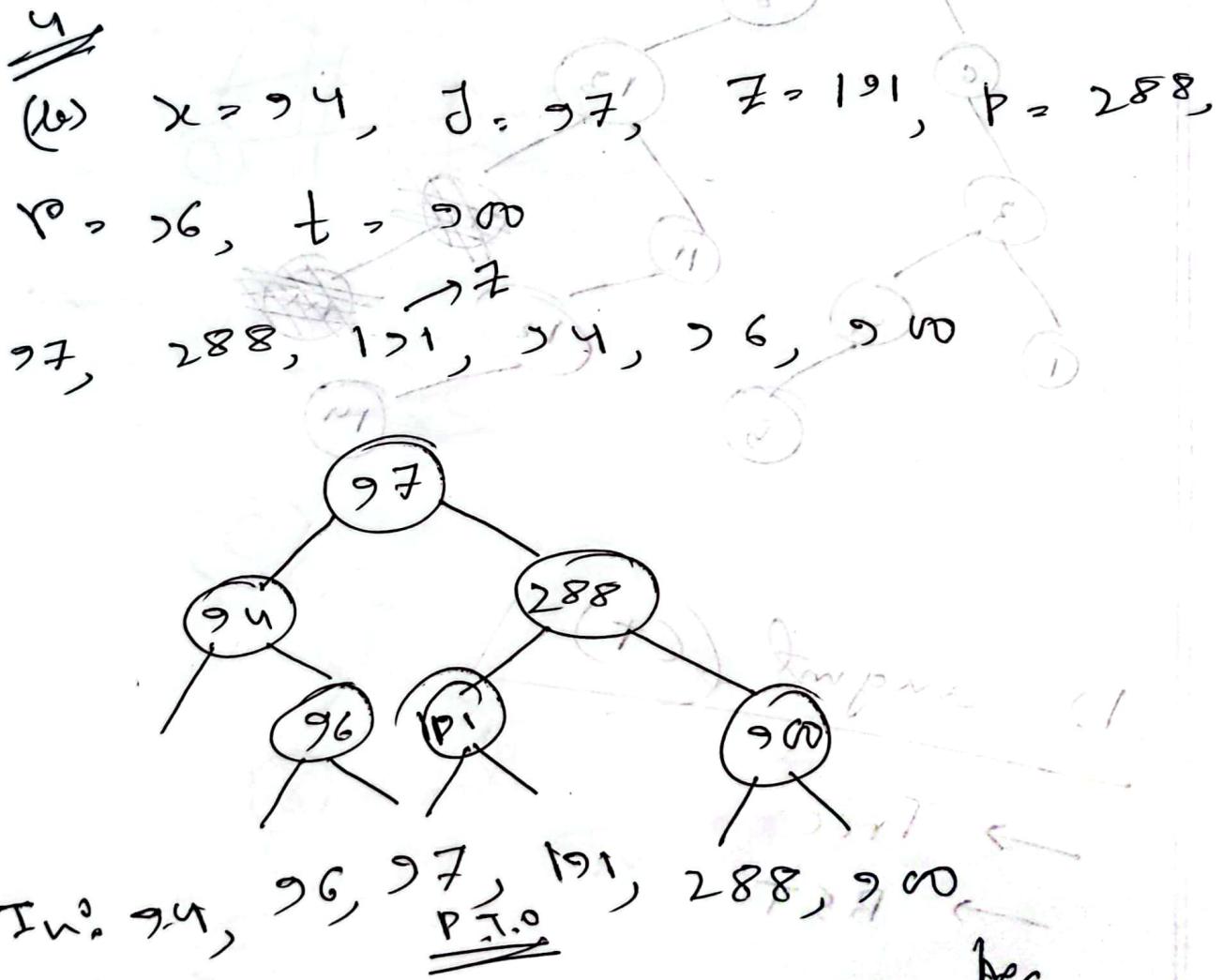
→ BST

right subtree \rightarrow
Inorder

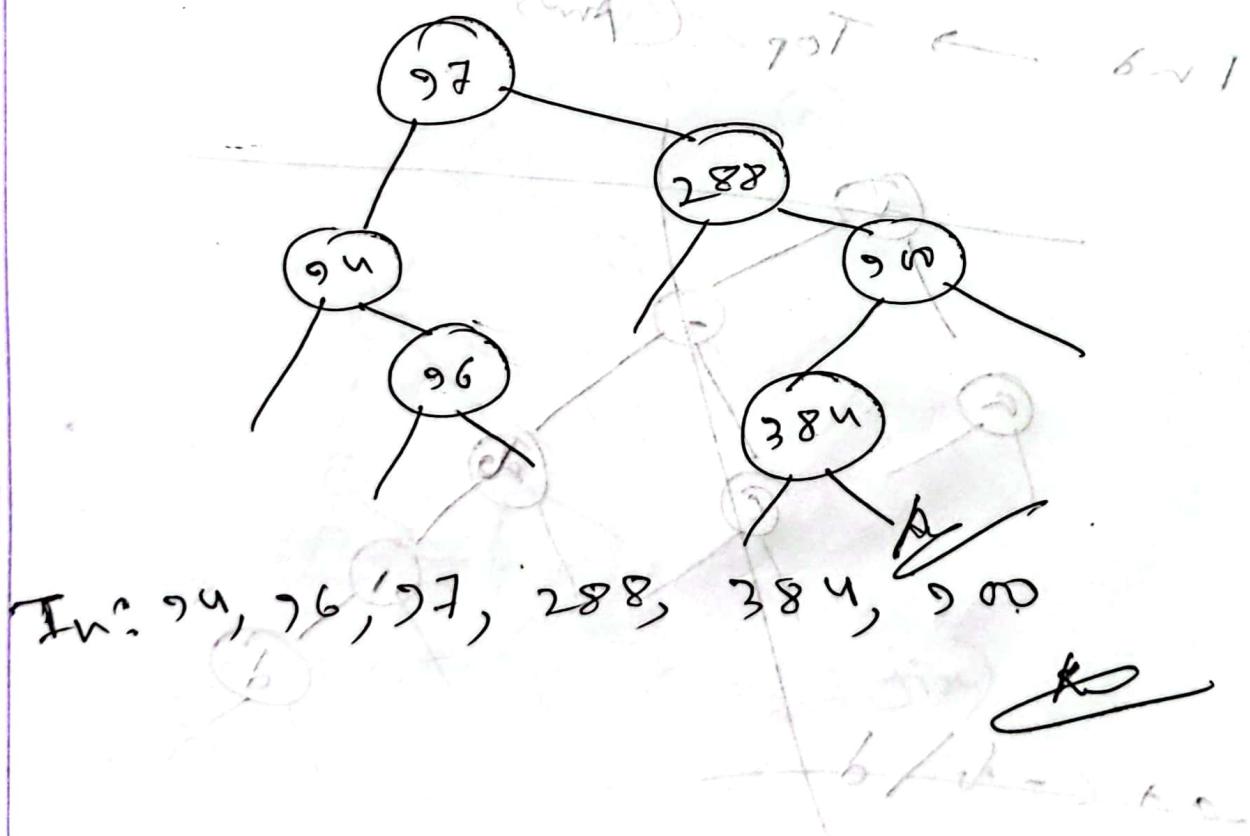
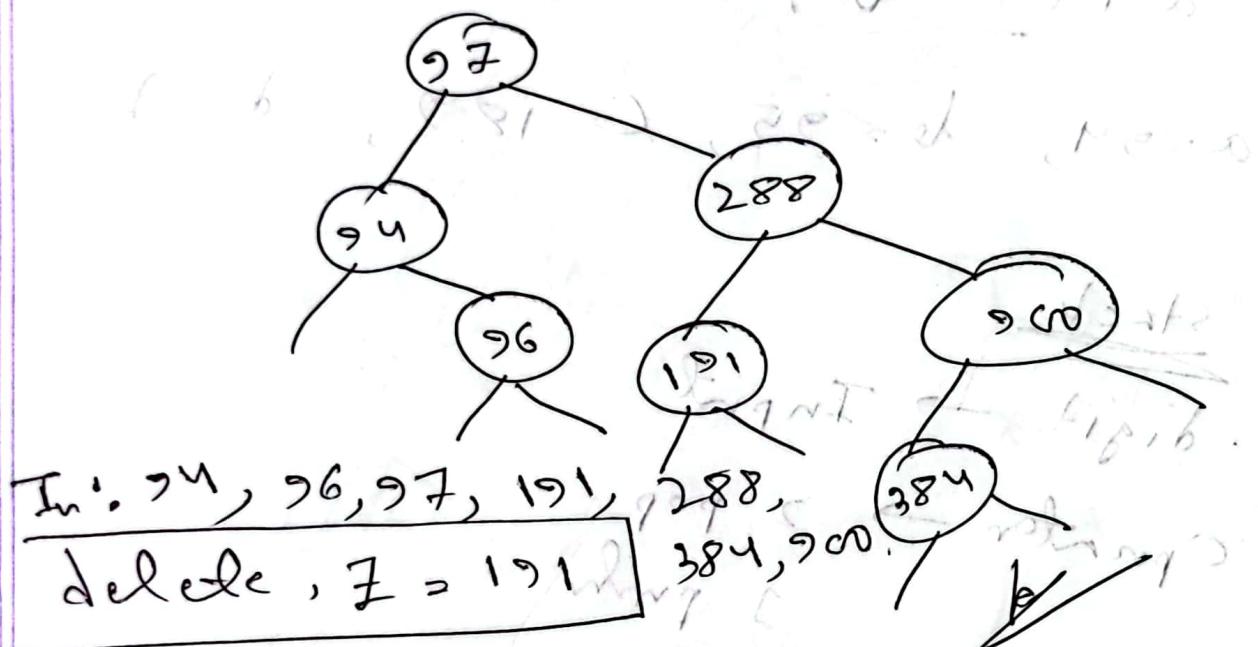
~~Successor \rightarrow Right subtree~~

~~Left child
Right child~~

~~Fall-22~~



insertion, $P+P = 384$



4(d)

$$a + \frac{(-b) + c}{d} \quad \text{(left root R)}$$

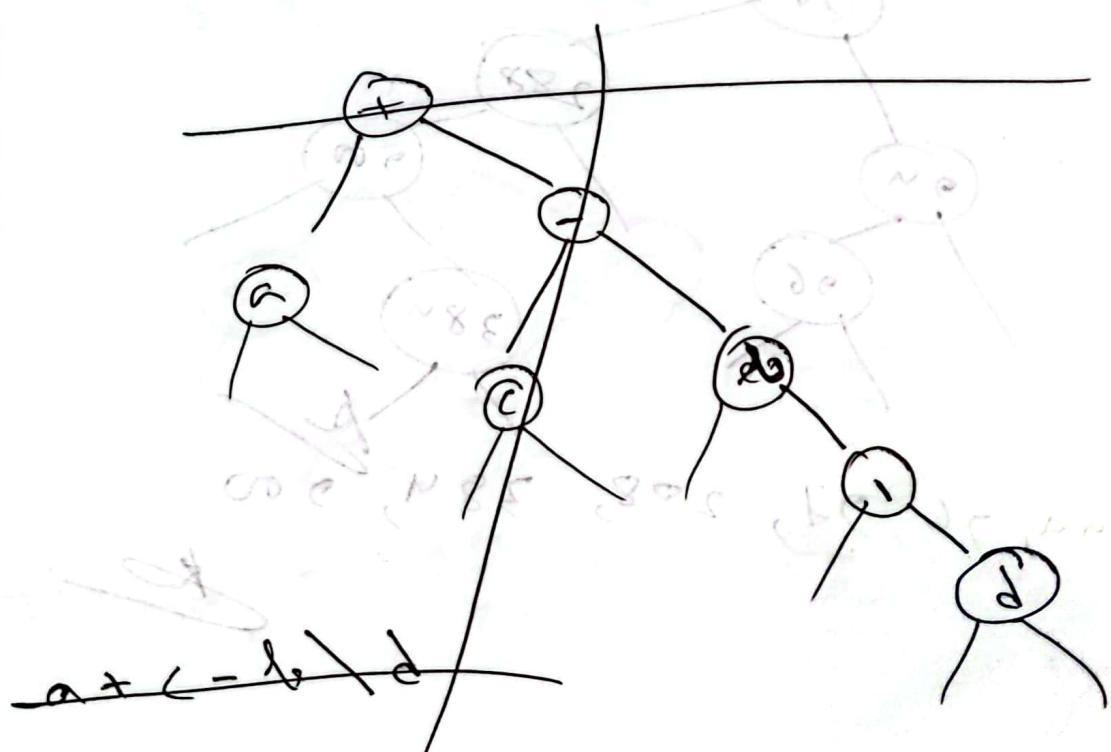
$$a = 94, b = 95, c = 189, d = 7$$

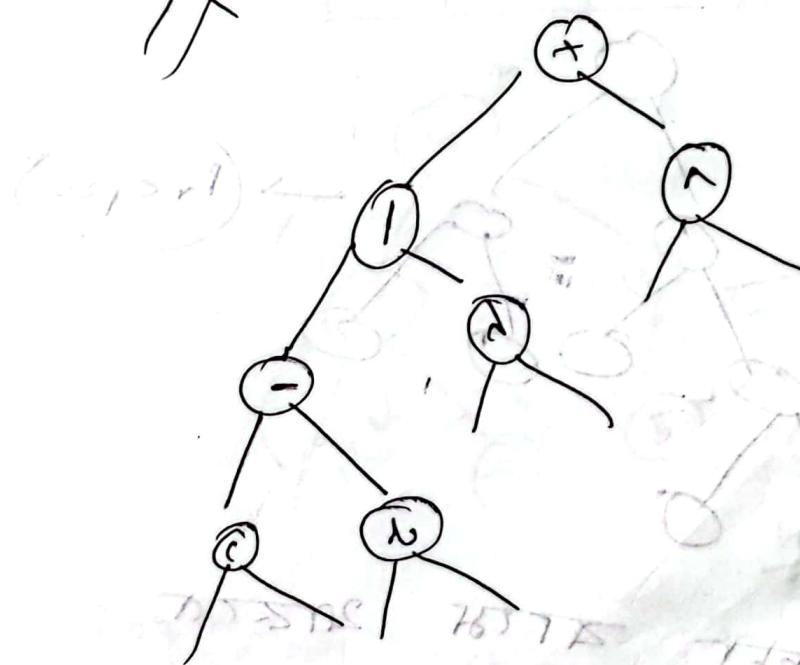
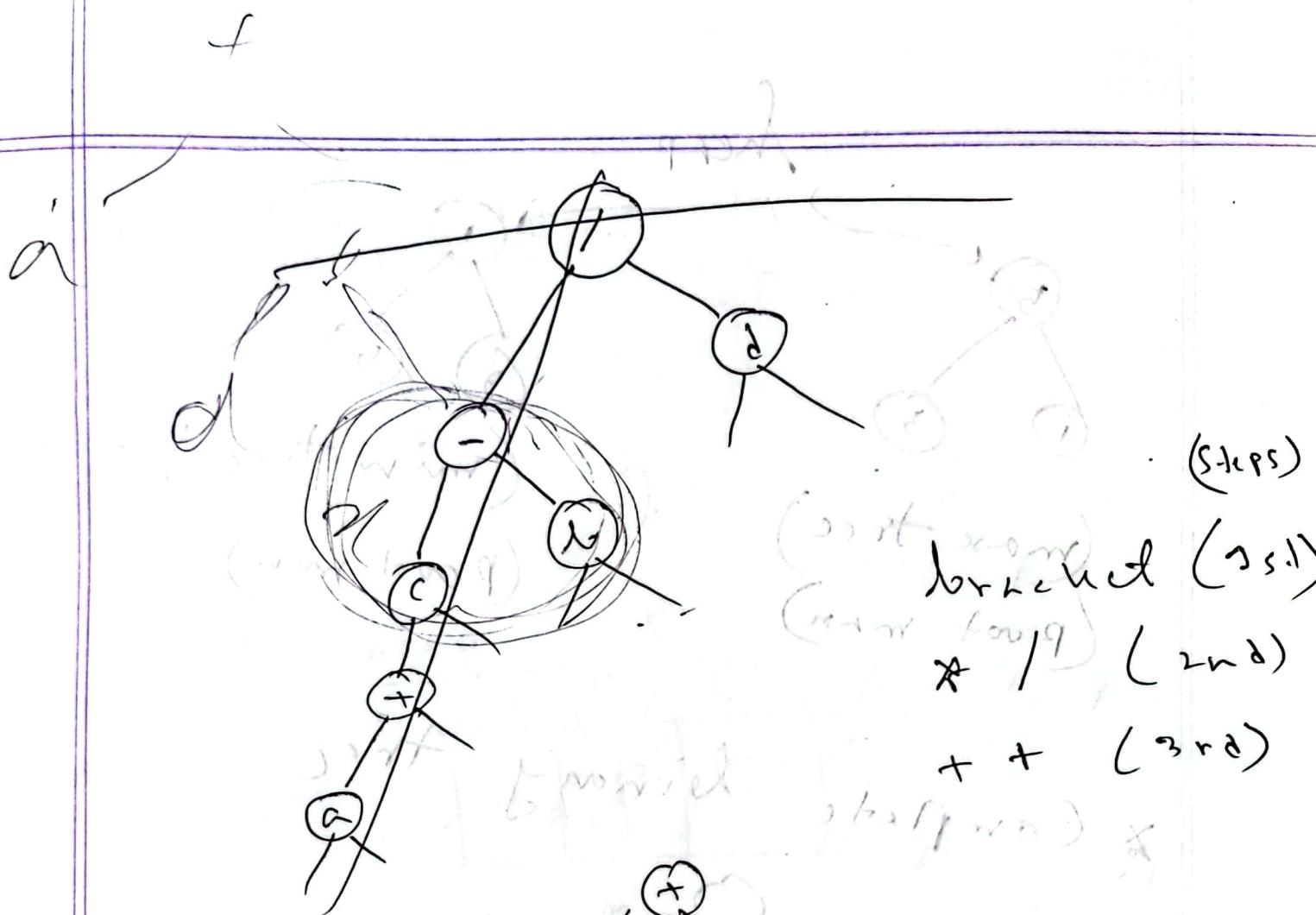
stack

digit → Input

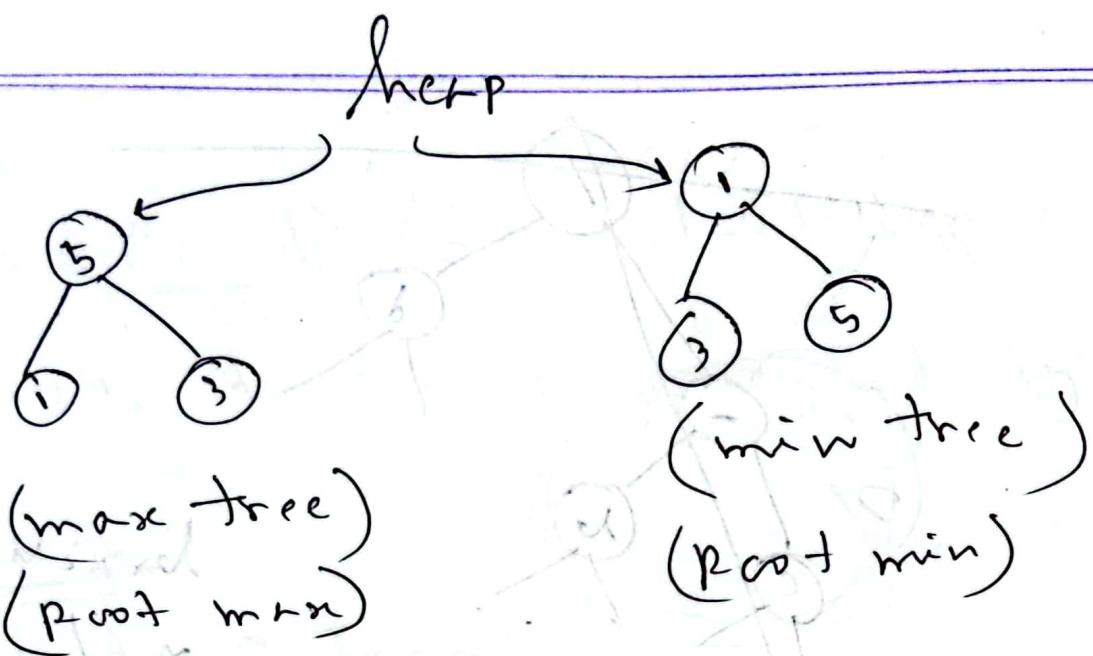
Operator → 2 pop
1 push

End → Top (Ans)

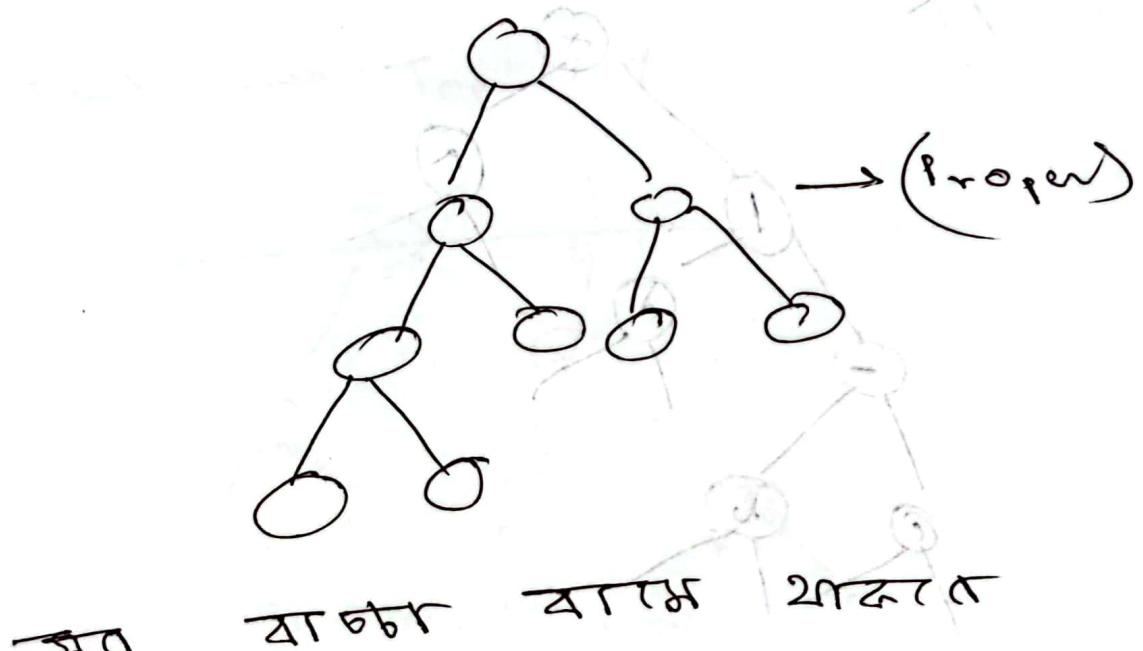




$b - d / a + c$ (Postfix)



* Complete binary tree



DB
BFS
DFS
(BFS) \rightarrow 6 nodes

~~2.1~~
8. 23
~~8.~~
Max-heap

$$h = \log n$$

$$h+1$$

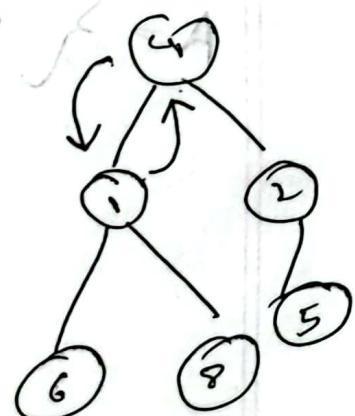
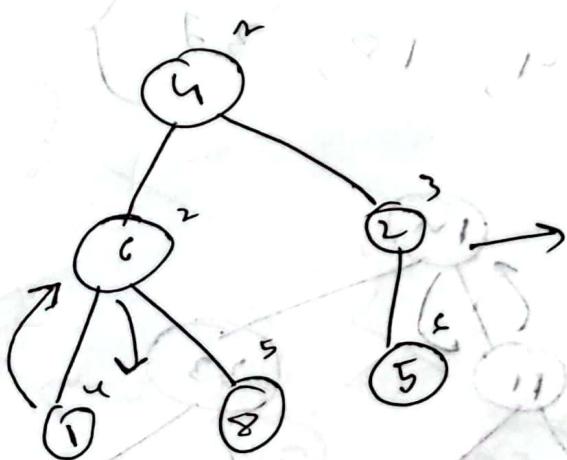
$$n = 2^h$$

~~8 Build-heap~~

0	1	2	3	4	5	6
6	4	6	2	1	8	5

size

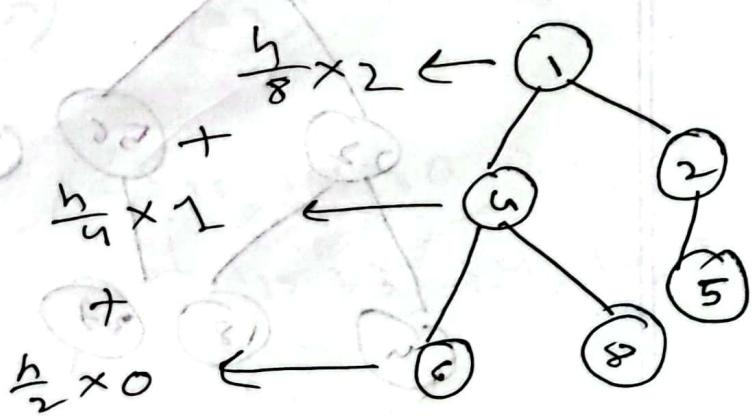
min

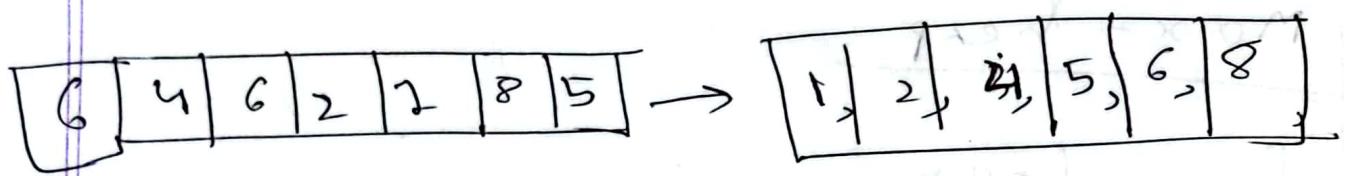


$$h = \log n$$

$$O(n \log n)$$

$$O(n)$$



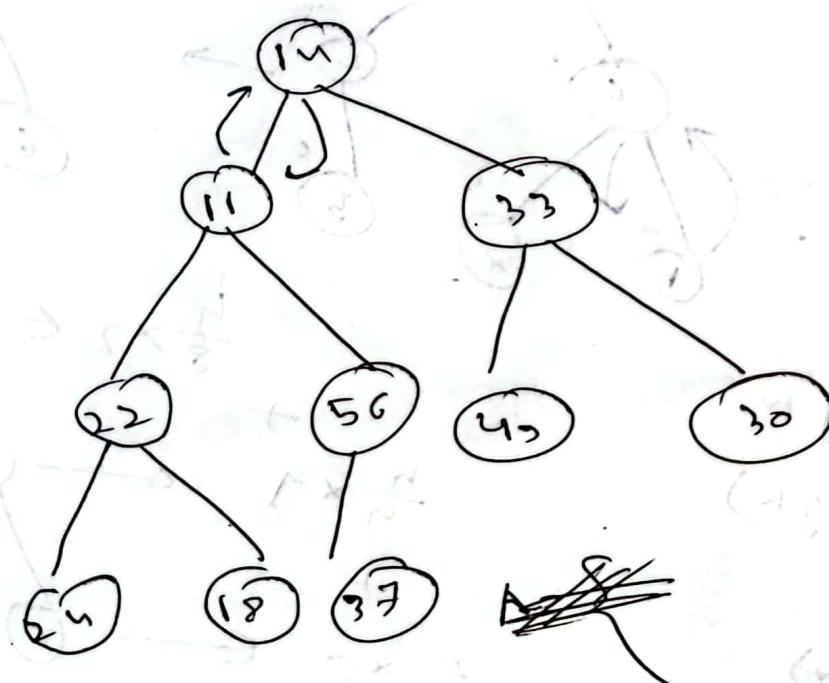


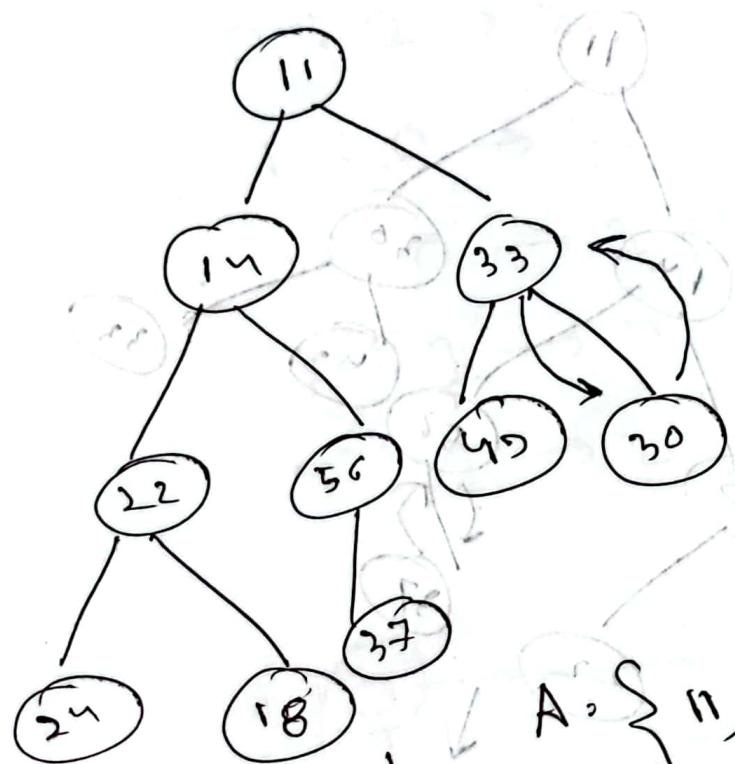
$$\frac{o(n)}{\downarrow} + \frac{o(n \log n)}{\downarrow}$$

build heapify

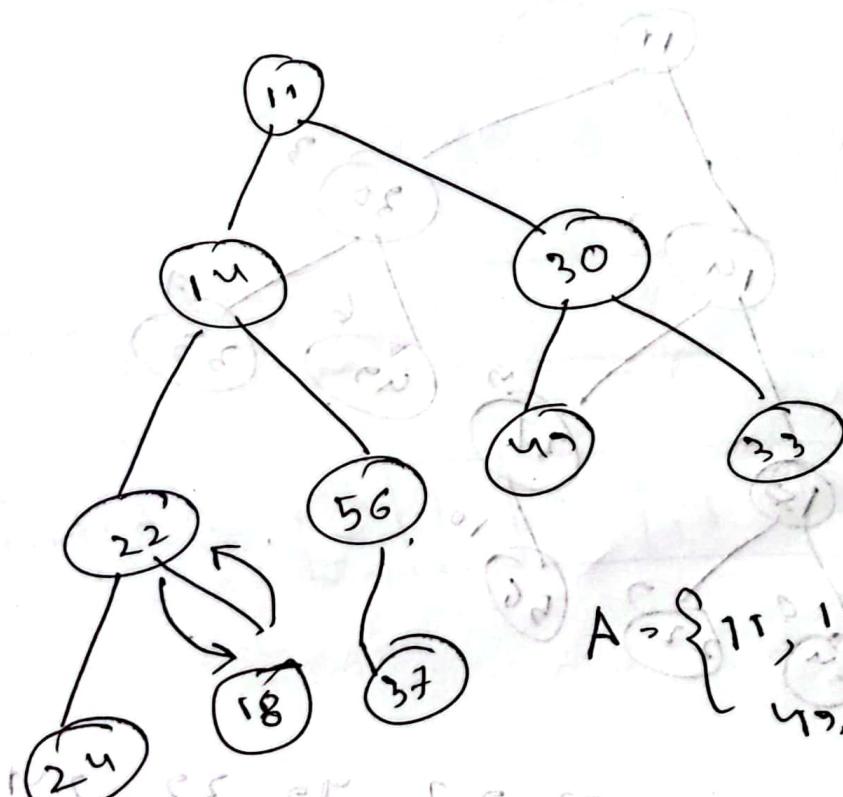
* Build \rightarrow min \rightarrow heap

$A = \{13, 11, 33, 22, 56, 49, 30,$
 $24, 18, 37\}$

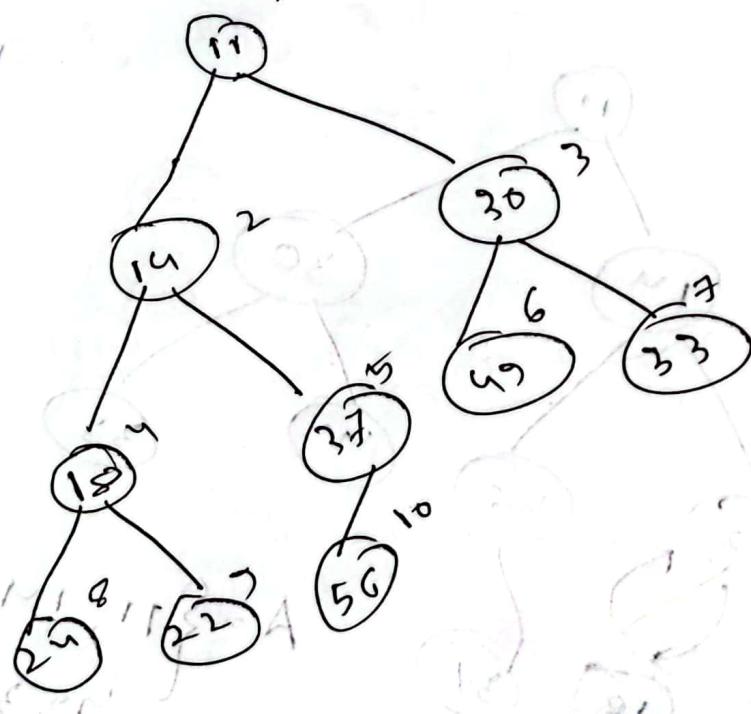
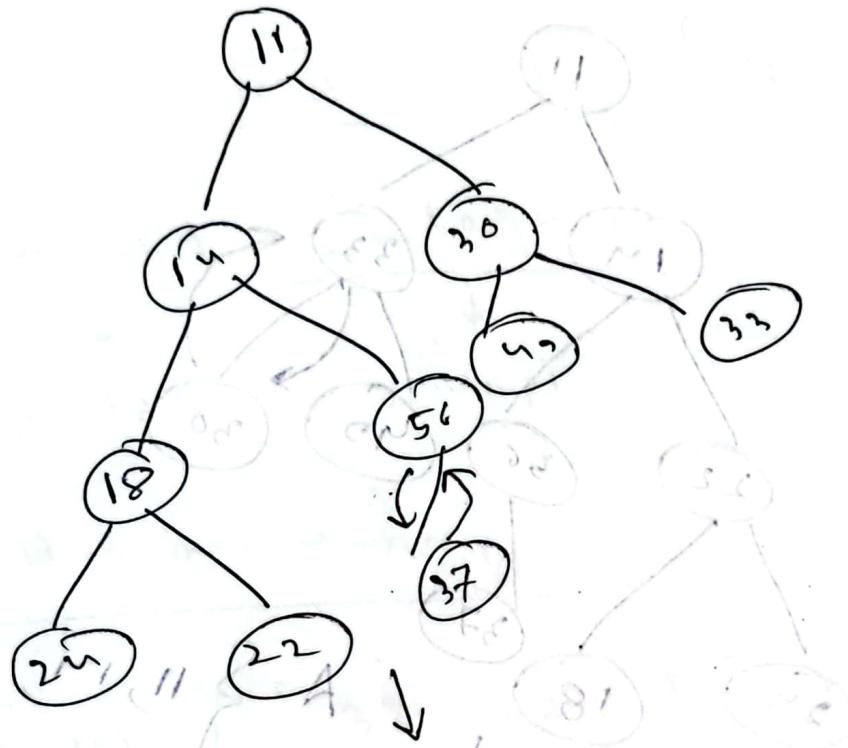
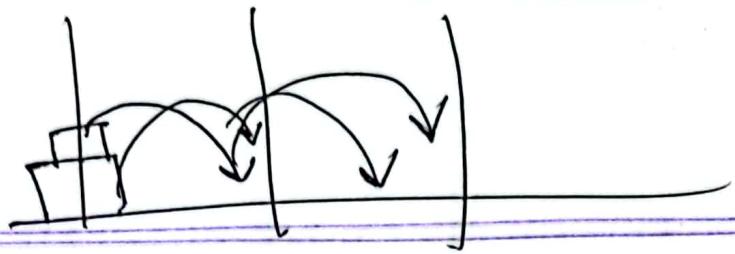




$A = \{11, 14, 33, 22, 56, 47, 30, 24, 18, 37\}$



$A = \{11, 14, 30, 22, 56, 47, 33, 24, 18, 37\}$



$A = \{11, 14, 30, 18, 37, 49, 33, 21, 22, 56\}$

BFS

(0, 1) ✓

(0, 2) ✓

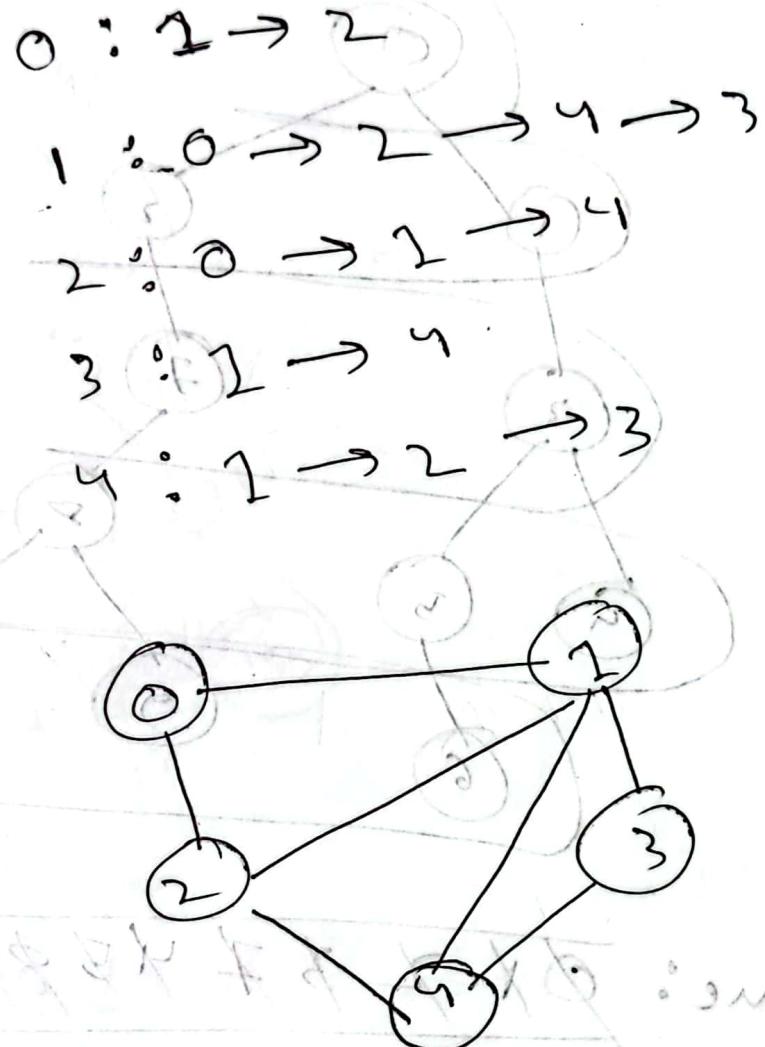
(1, 2) ✓

(1, 4) ✓

(1, 3) ✓

(2, 4) ✓

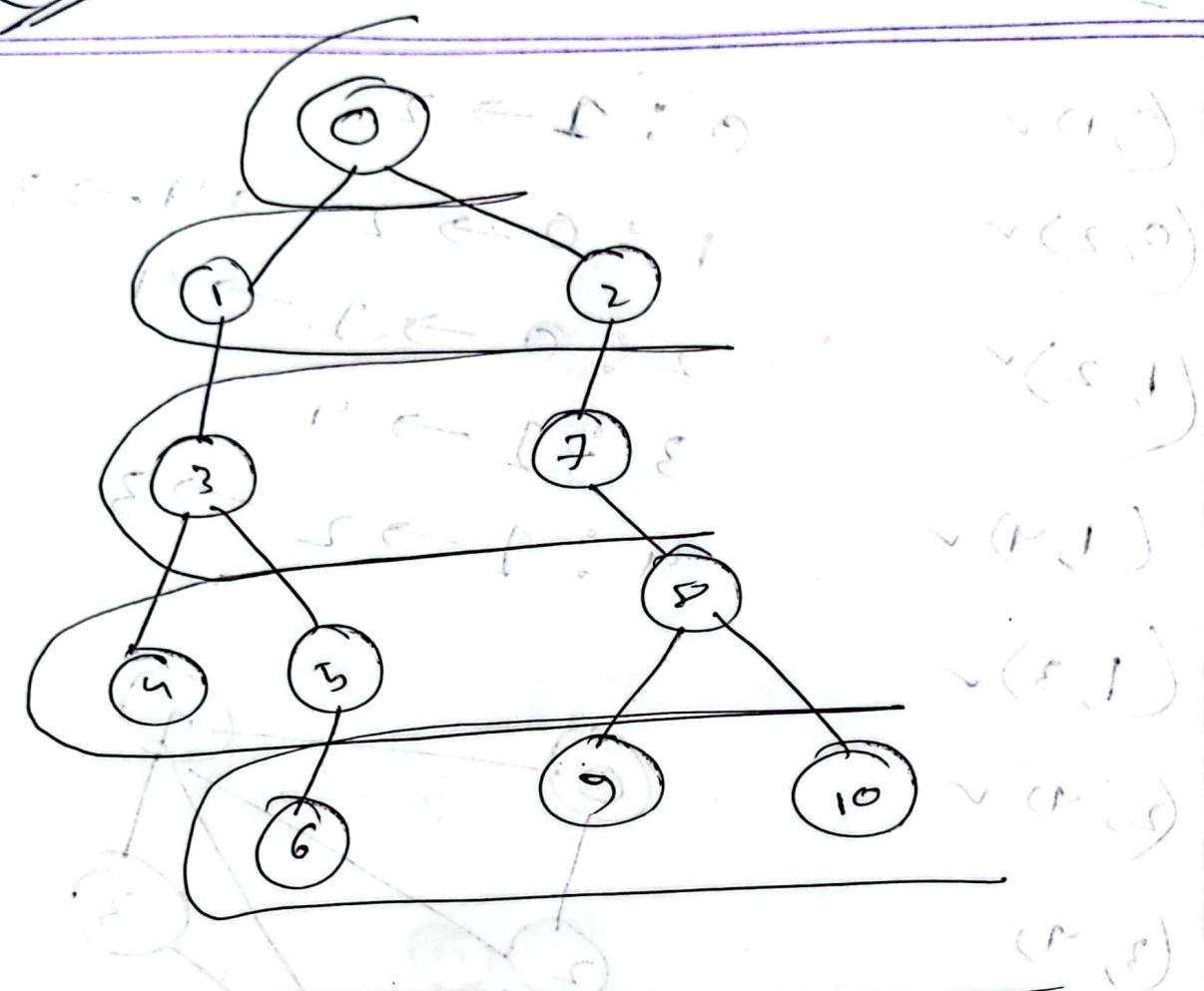
(3, 4)



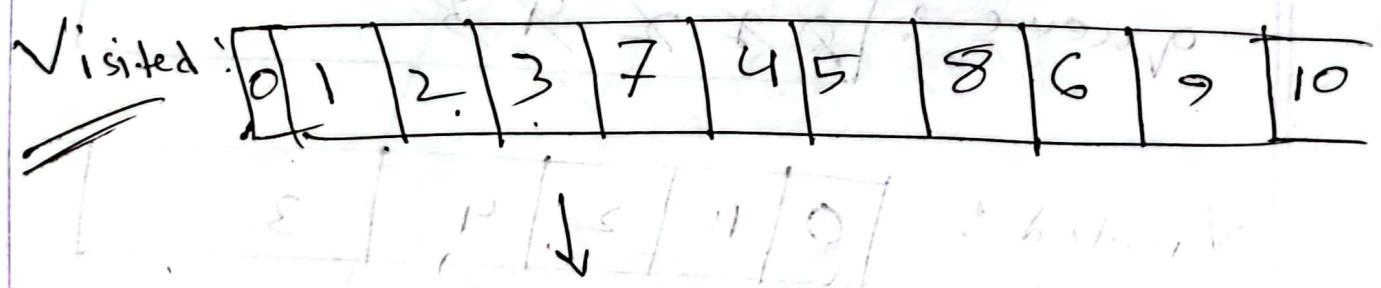
queue : [0 2 1 4 3]

visited : [0 1 2 4 3]

~~BFS \rightarrow queue~~

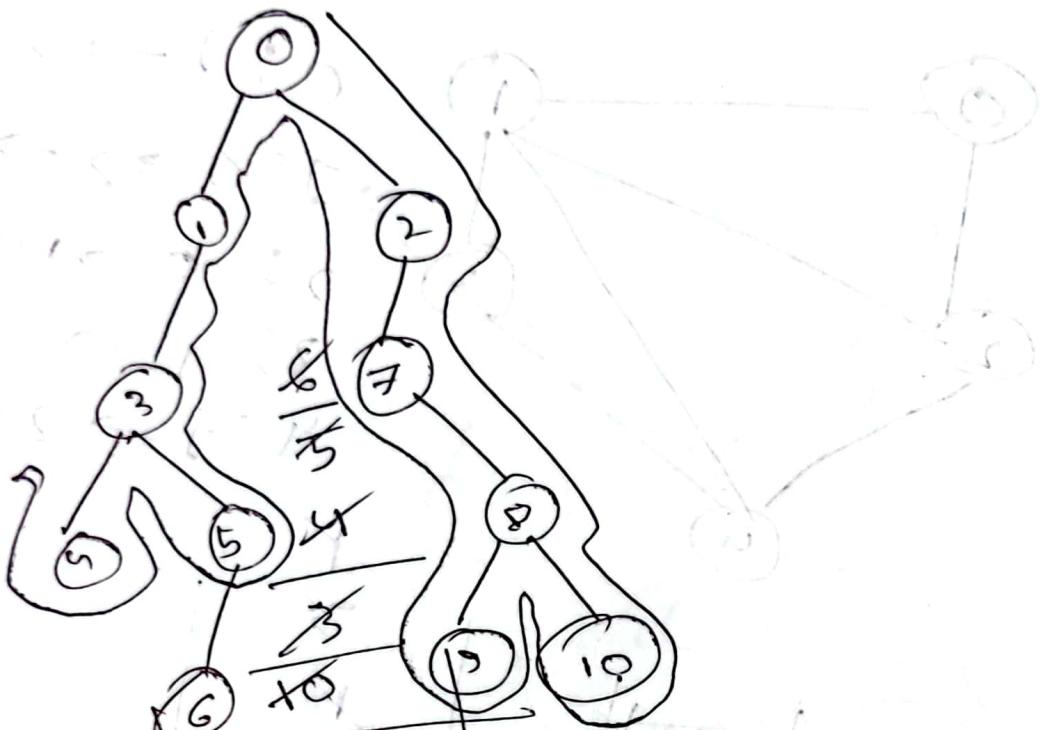


Ans: ~~0 X-2 3 7 4 5 8 6 9 10~~

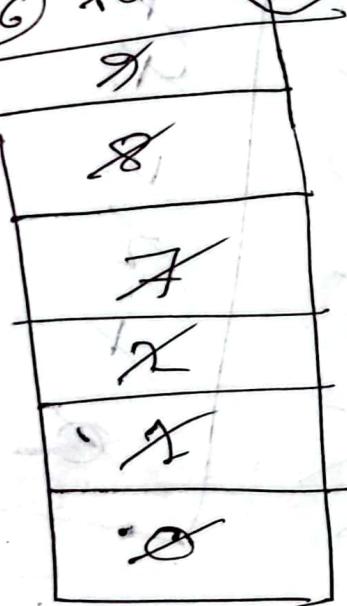


level order traversal

DFS → (stack)



stack:

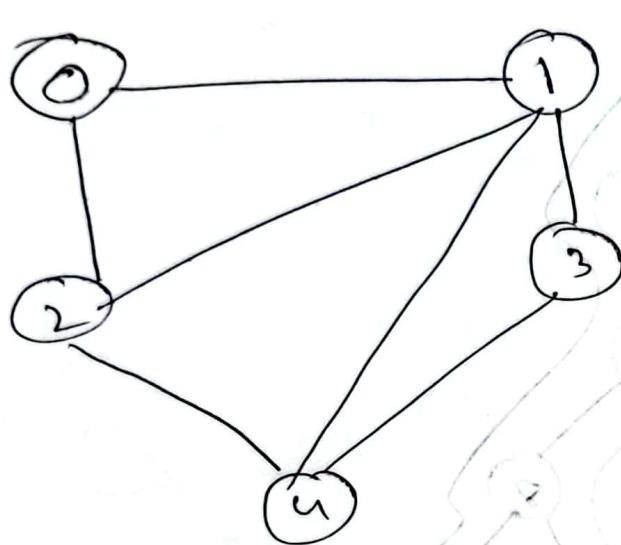


Visited:

0	2	7	8	10	9	1	3	5, 6	4
---	---	---	---	----	---	---	---	------	---

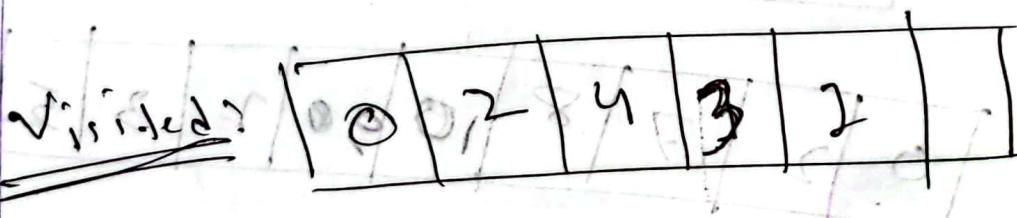
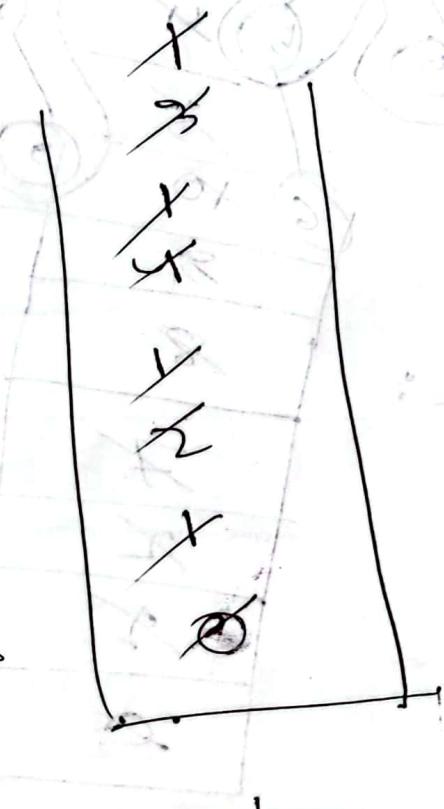
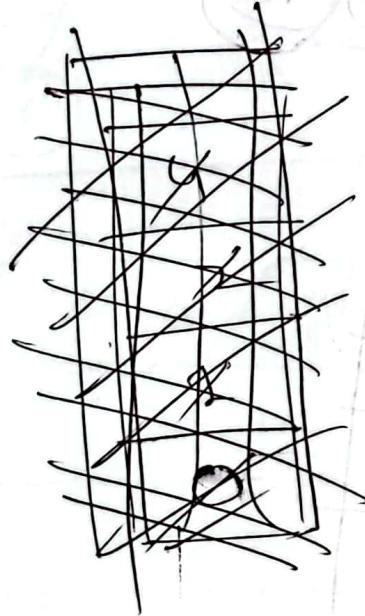


depth order traversal



$0: 2 \rightarrow 2$
 $1: 0 \rightarrow 2 \rightarrow 4 \rightarrow 3$
 $2: 0 \rightarrow 1 \rightarrow 4$
 $3: 2 \rightarrow 4$
 $4: 2 \rightarrow 2 \rightarrow 3$

stack:

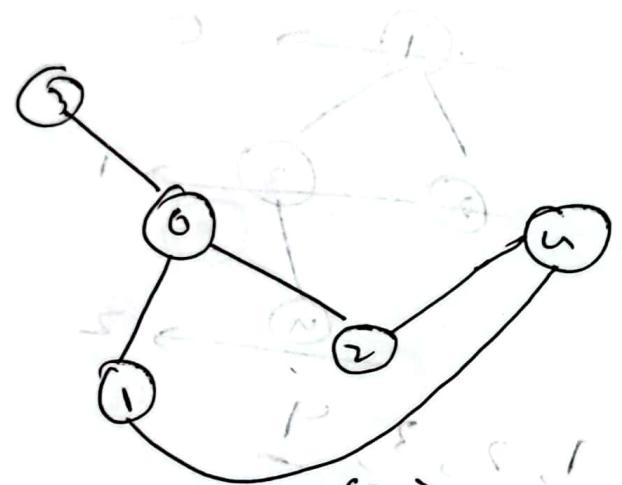


Visited nodes: ↴

LW
19.8.23

DSA 2

Graph



0: - 3 → 1 → 2

1: 0 → 4

2: 0 → 4

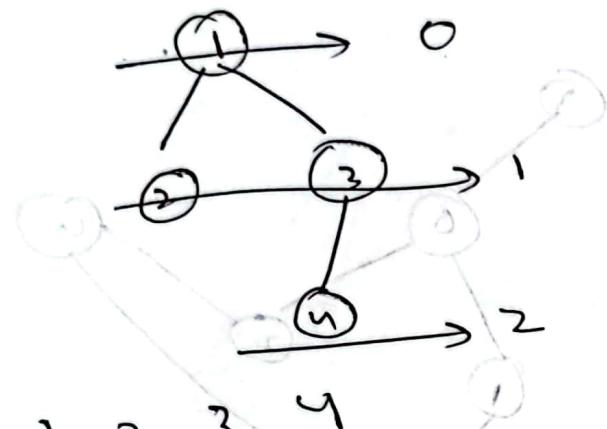
3: 0 → 4

4: 2 → 3 → 4

$O(n)$ ~~adjacency list~~

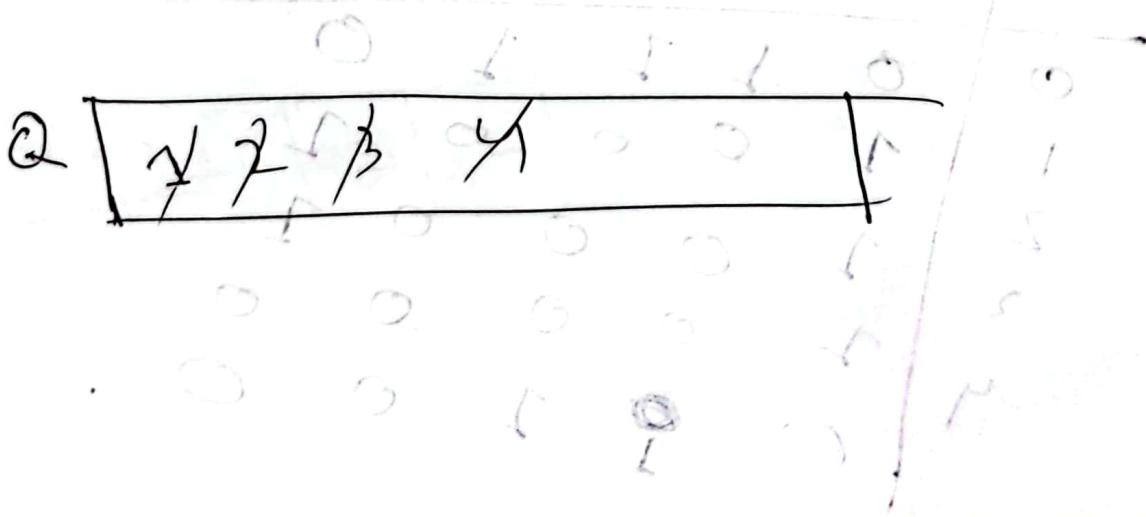
	0	1	2	3	4
0	0	1	1	1	0
1	1	0	0	0	1
2	1	0	0	0	1
3	1	0	0	0	0
4	0	1	1	0	0

BFS

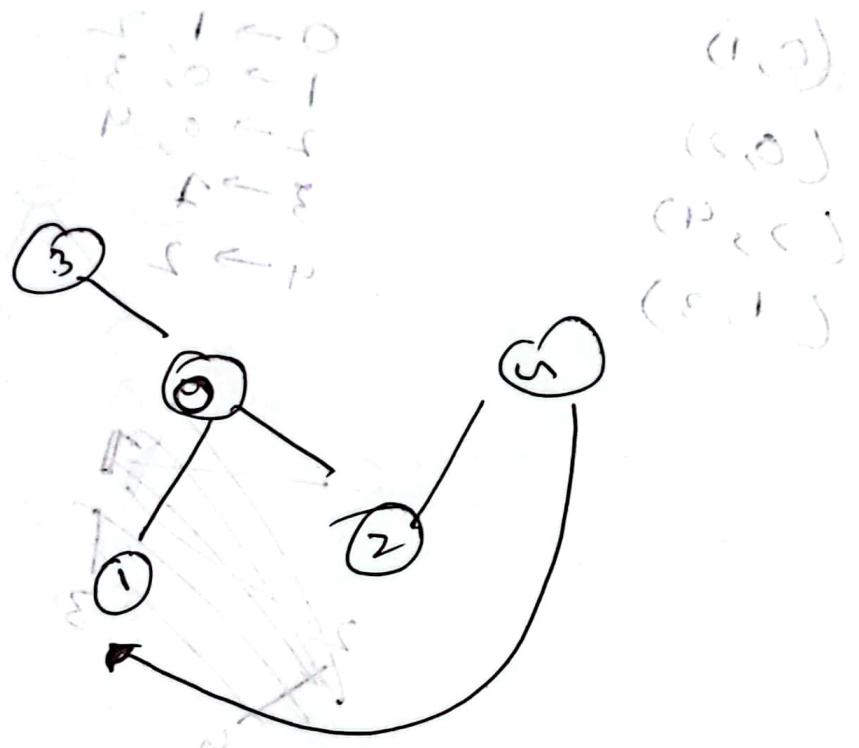


Visit : 1, 2, 3, 4

	0	1	2	3	4	
(1, 2)	0	0	0	0	0	
(1, 3)	1	0	0	1	2	
(2, 3)	2	0	1	0	0	
(3, 4)	3	0	1	0	0	
(4, 5)	4	0	0	0	1	

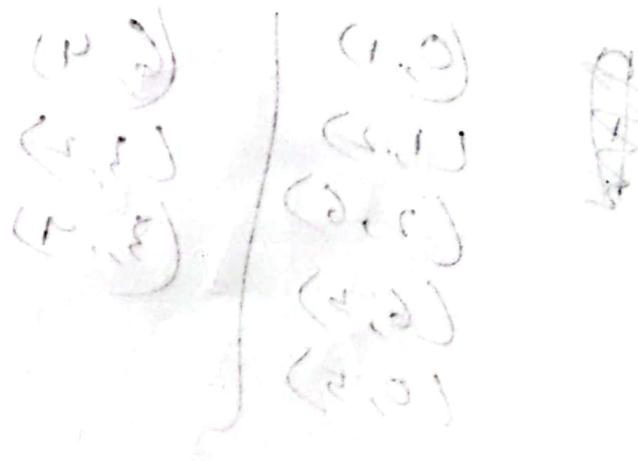


DFS



S XØ X₂/₃

Visited 14₂₀₃

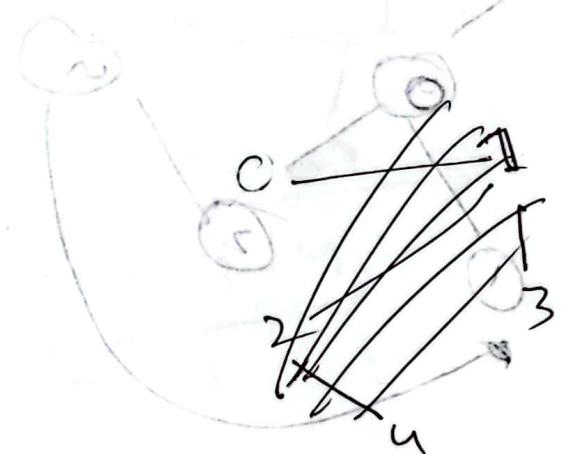


$(0, 1)$
 $(0, 2)$
 $(2, 4)$
 $(1, 3)$

$0 \rightarrow 1, 2$
 $1 \rightarrow 0, 3$
 $2 \rightarrow 0, 4$
 $3 \rightarrow 1$
 $4 \rightarrow 2$

270

~~270~~



$0 \rightarrow 1$
 $1 \rightarrow 2$
 $2 \rightarrow 5$
 $5 \rightarrow 6$

$0 \rightarrow 1$
 $1 \rightarrow 3$
 $3 \rightarrow 4$
 $4 \rightarrow 5$
 $5 \rightarrow 6$

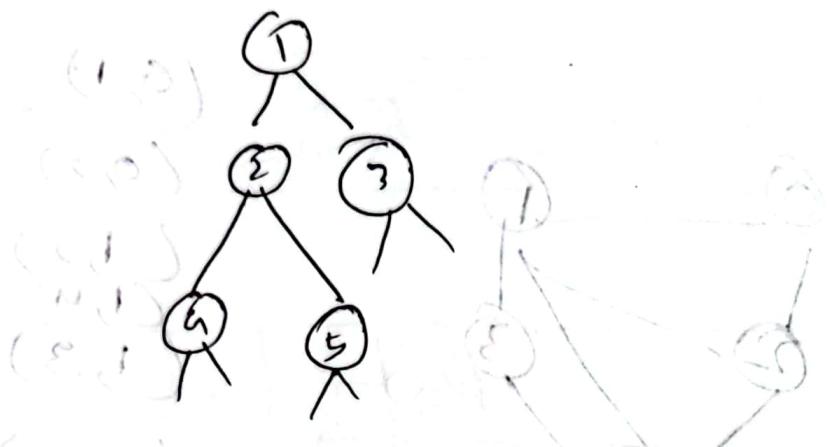
$0: 1, 2$
 $1: 3, 0$
 $2: 0, 4$

$3: 1$
 $4: 2$

with v

~~0, 1~~
 $(0, 1)$
 $(1, 2)$
 $(2, 6)$
 $(6, 5)$
 $(0, 5)$

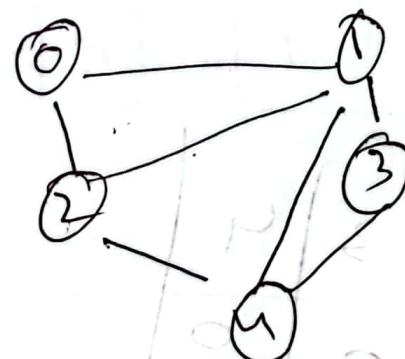
$(6, 4)$
 $(3, 2)$
 $(3, 1)$



Pre: 1, 2, 4, 5, 3

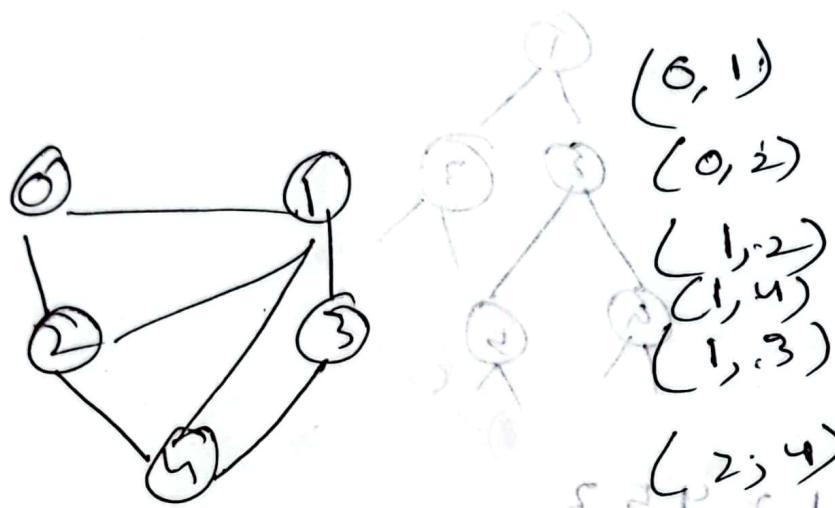
In: 4, 2, 5, 1, 3

Post: 4, 5, 2, 3, 1



	S	I	O	R	T	E
L	1	0	1	0	1	0
F	0	0	0	1	1	0
R	0	0	1	0	0	1
O	1	1	1	1	0	1

DFS



0 : 1 → 2

1 : 0 → 2 → 4 → 3

2 : 0 → 1 → 4

3 : 1 → 4

4 : 1 → 2 → 3

$\begin{matrix} & & & & \\ \mathcal{E}_1 & \mathcal{E}_2 & \mathcal{E}_3 & \mathcal{E}_4 & \dots \\ \mathcal{E}_1 & \mathcal{E}_2 & \mathcal{E}_3 & \mathcal{E}_4 & \dots \\ \mathcal{E}_1 & \mathcal{E}_2 & \mathcal{E}_3 & \mathcal{E}_4 & \dots \\ \mathcal{E}_1 & \mathcal{E}_2 & \mathcal{E}_3 & \mathcal{E}_4 & \dots \end{matrix}$

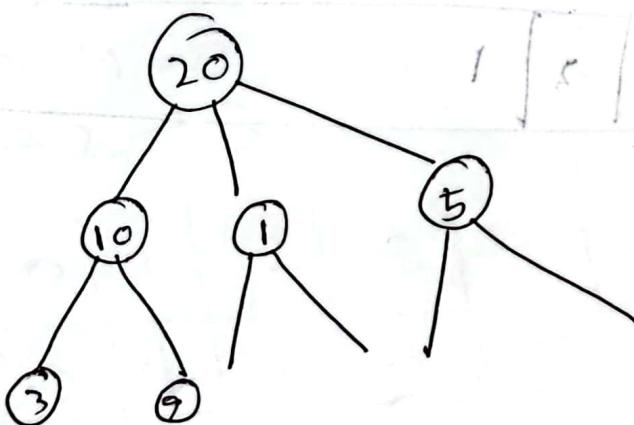
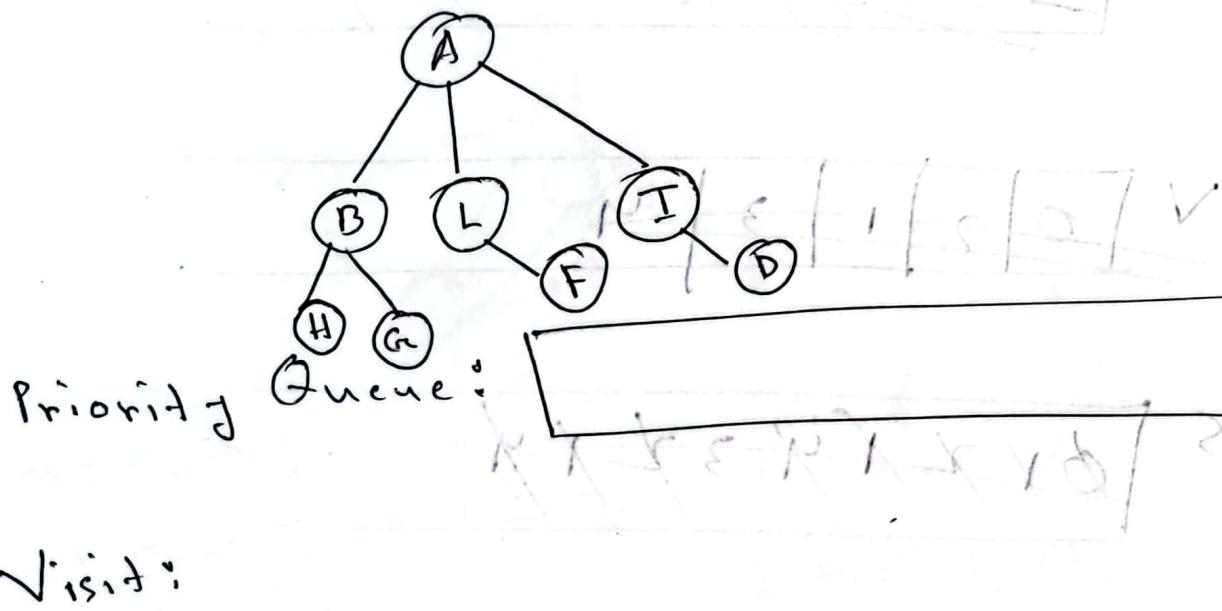
1 → 0, 1, 4, 3 → 9

	0	1	2	3	4	
0	0	1	2	0	0	
1	1	0	1	1	1	
2	1	1	0	0	1	
3	0	1	0	0	1	
4	0	1	1	1	0	

C.W
21.8.23

DS A1

BFS

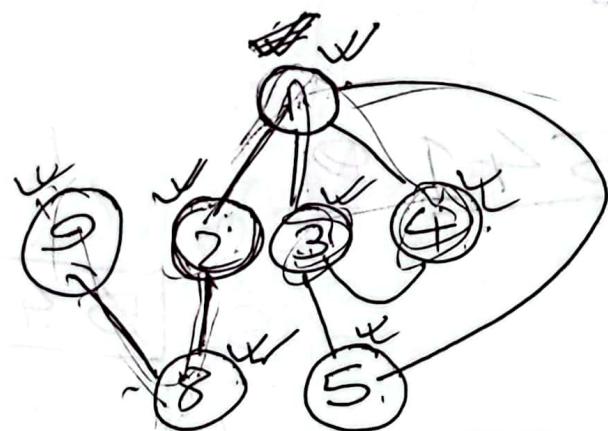


BFS: 20, 10, 1, 5, 3, 9

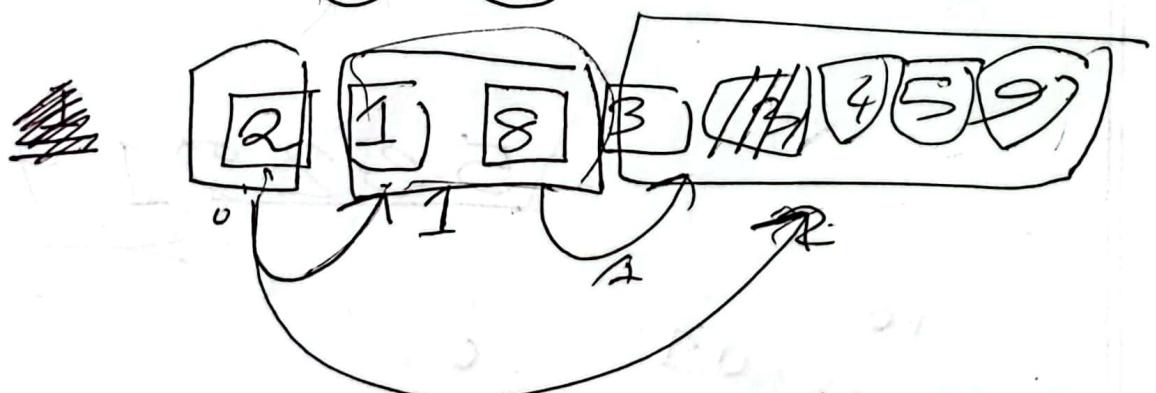
Min Priority

Priority: 20, 10, 9, 8, 5, 3, 1

BFS

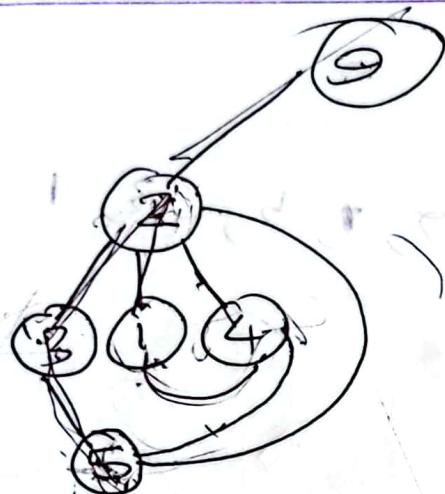


que

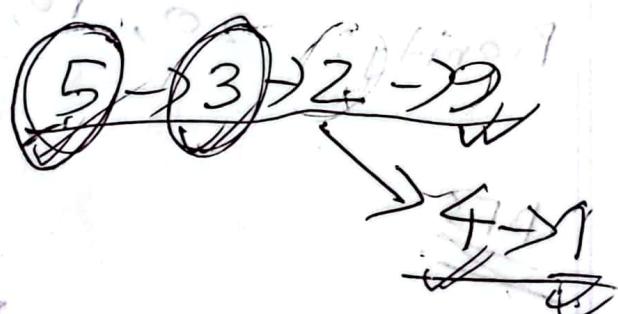


কাস্টম নোড মেরে দৃশ্যমান visit কর
কাস্টম তেরো কোর্স নোড ক

DFS



Visited set



2 3 5 4 1 9

S 5 4 3 2 1 9

5 3 2 9 1 4

Top first ~~top to bottom~~
Right most first ~~right to left~~

L-A2A

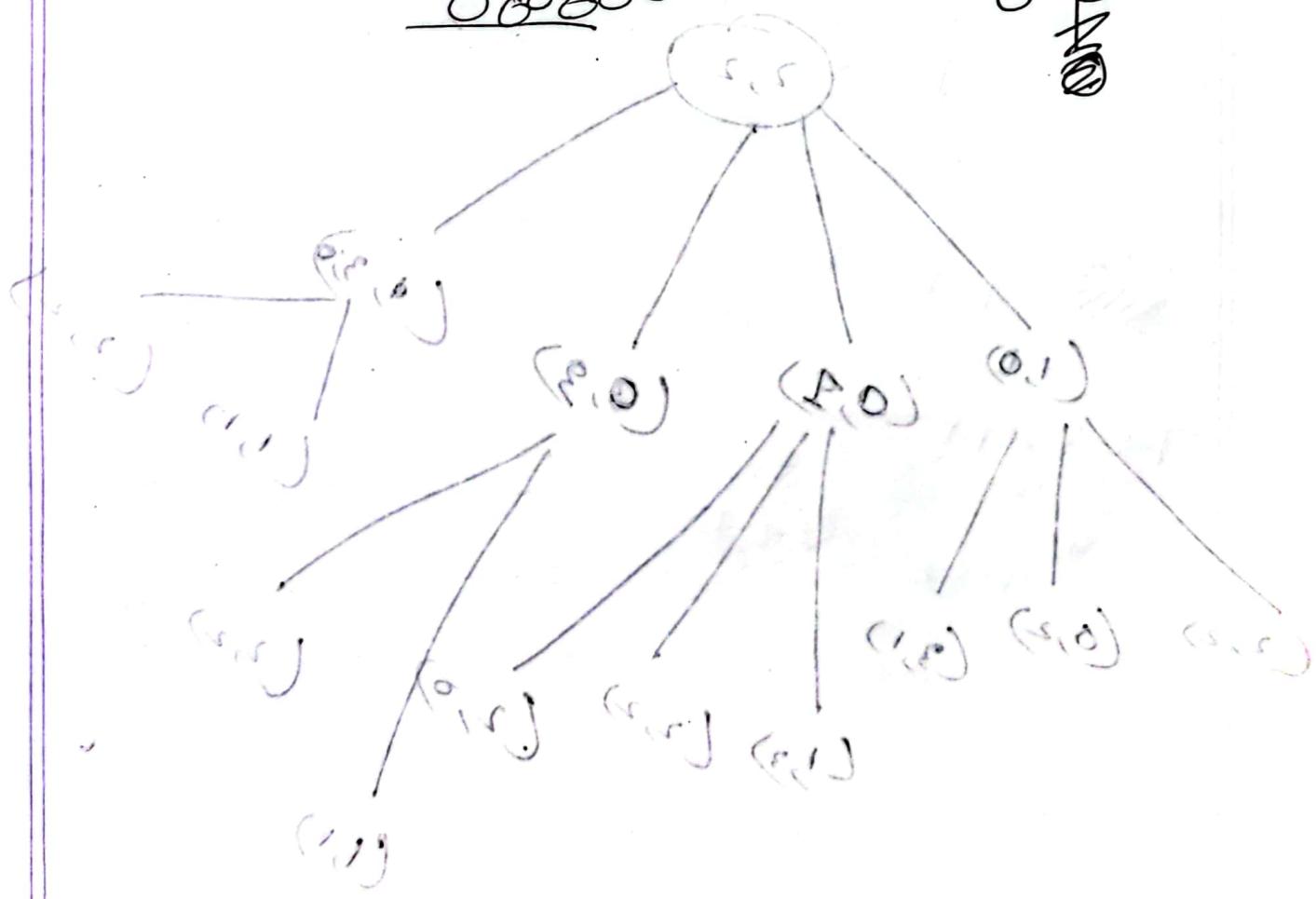
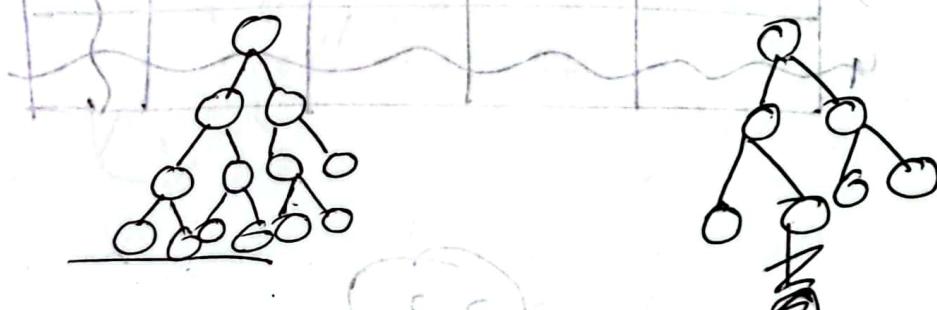
~~tom~~ Binary Tree: $(0, 1, 2)$

Proper BT \Rightarrow Child $(0, 2)$

~~tom~~ Complete: Internal node \Rightarrow 2

last level leaf node: $[0, 1, 2]$

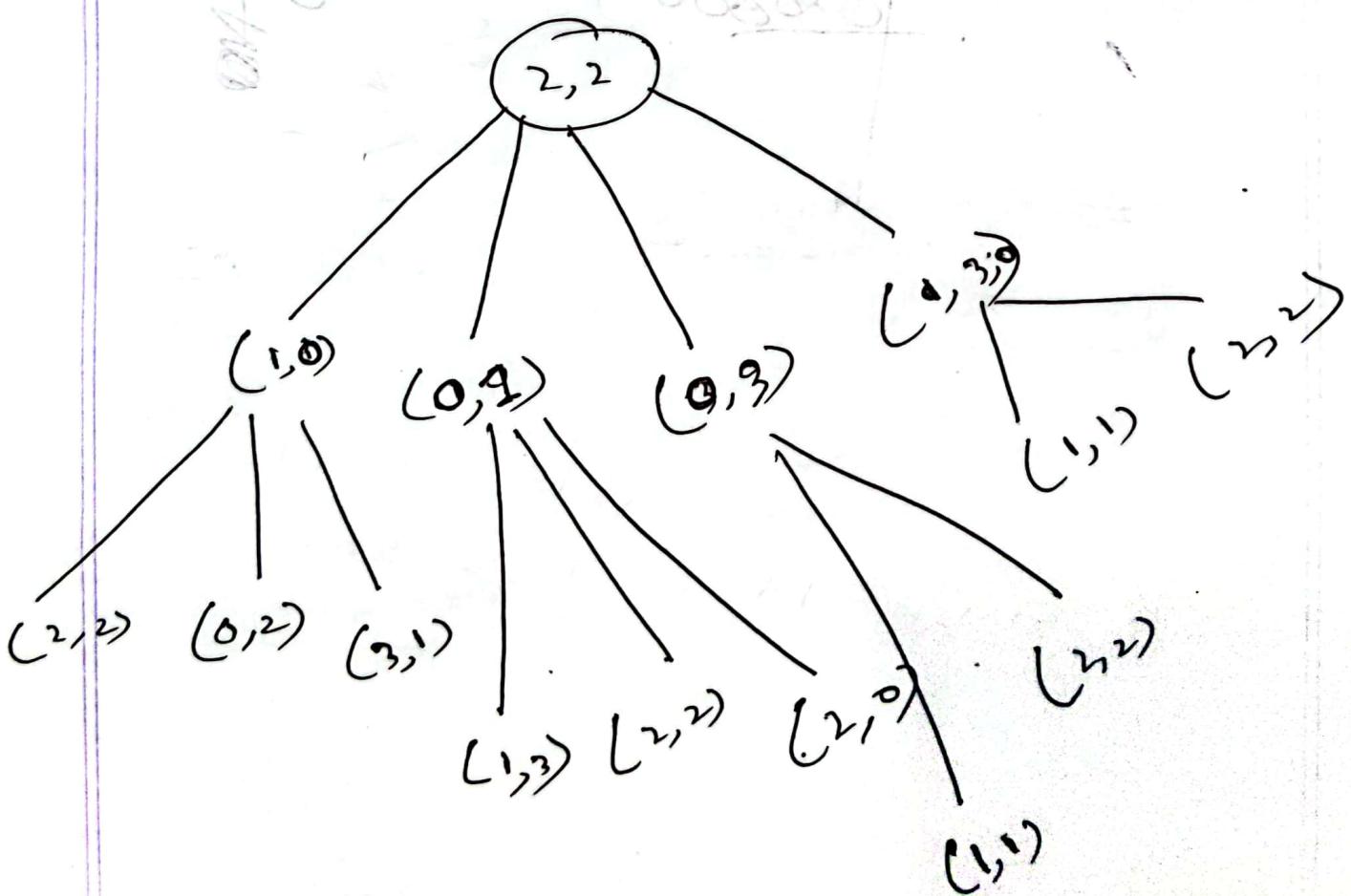
$2 \rightarrow$ Left Bias

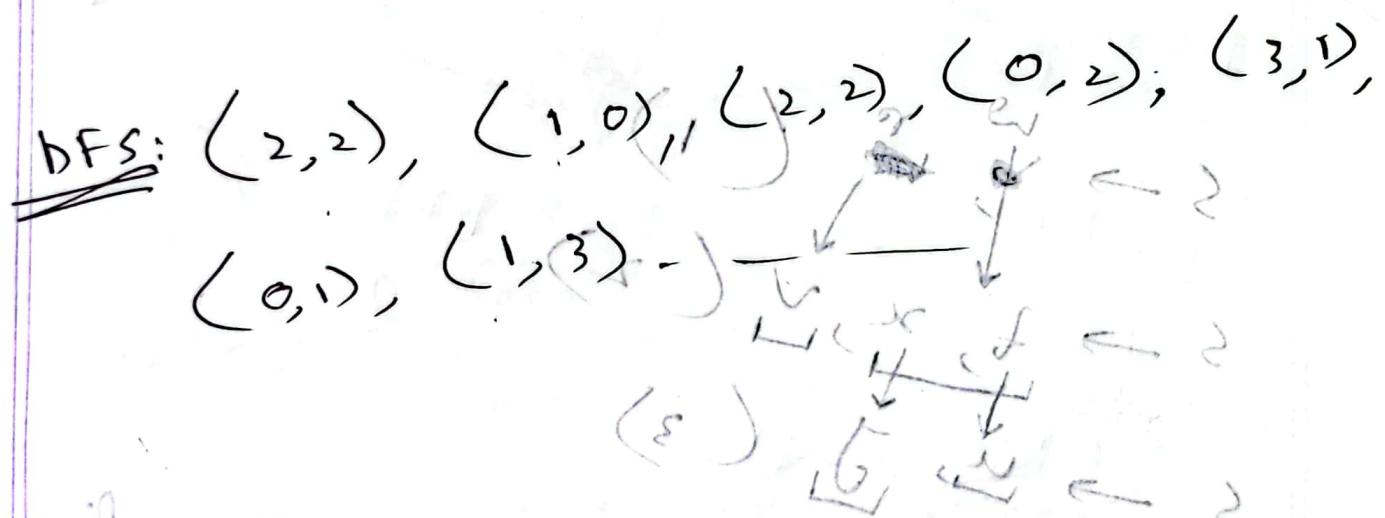
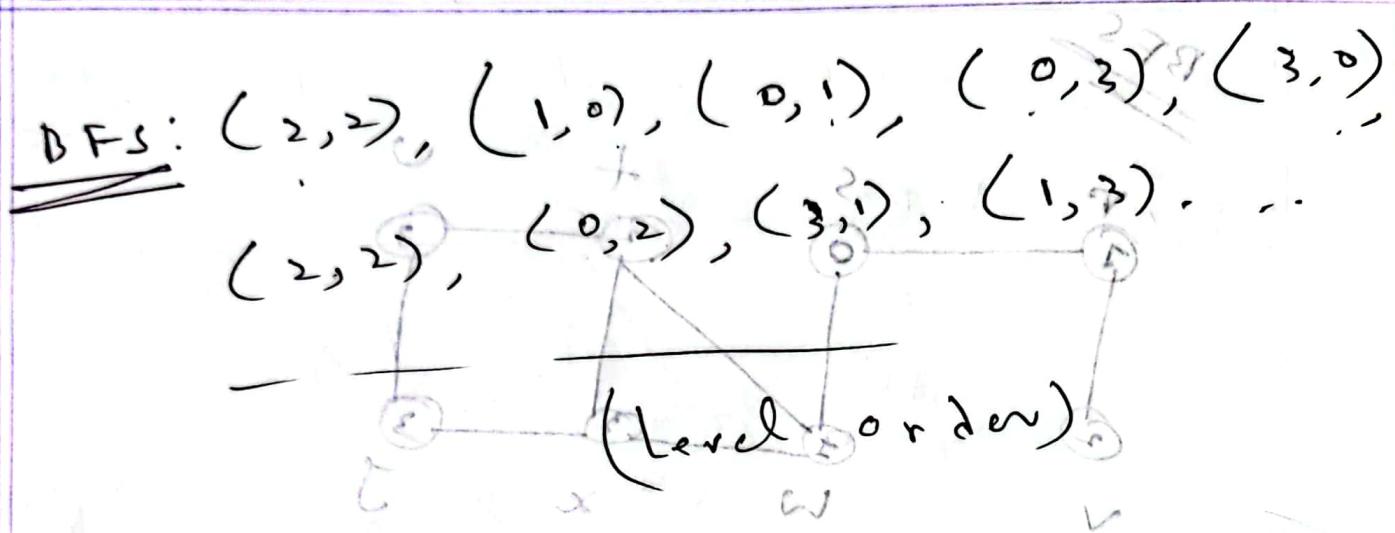


6/2
22.8.23

DSA-1

0	1	2	3	4
0	✓			
1	✓			
2		✓		
3	✓			
4				



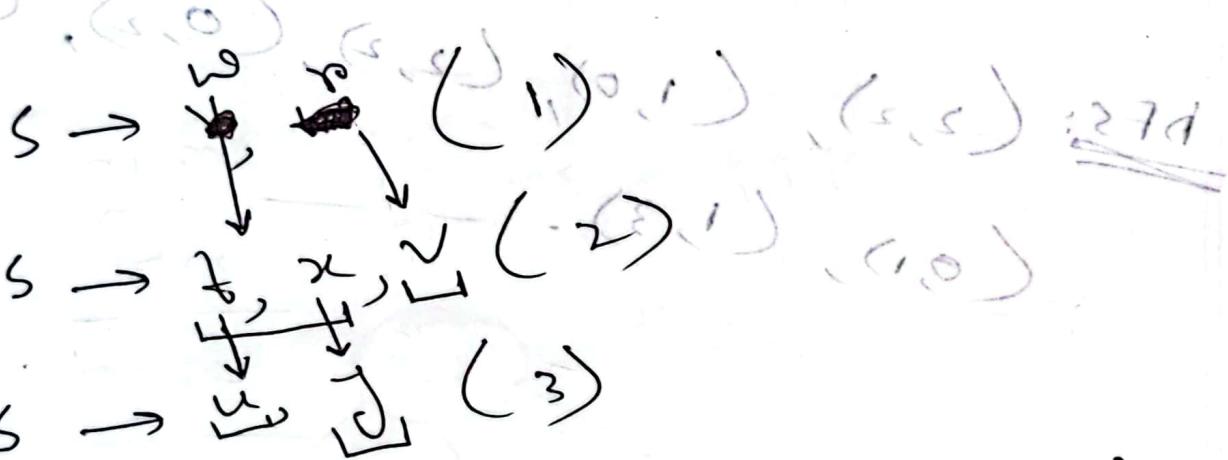
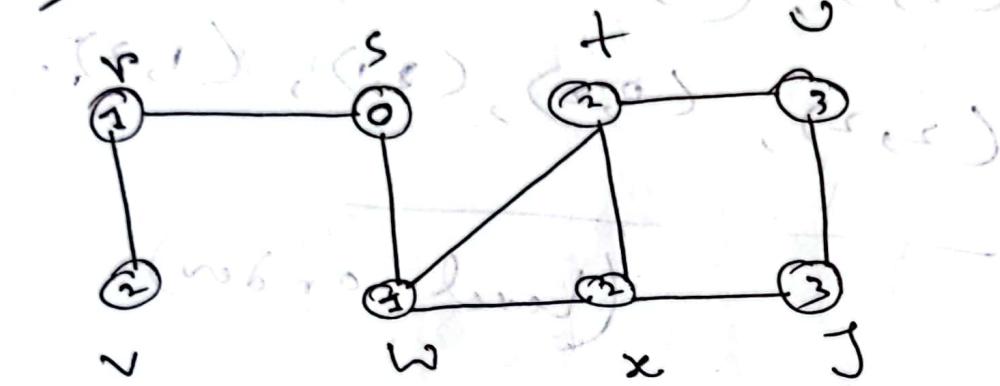


problems of strings + 277 ~~and~~

$$277 \quad | \quad 278$$

$$(3+2) \cdot 16 = 37$$

~~BFS~~

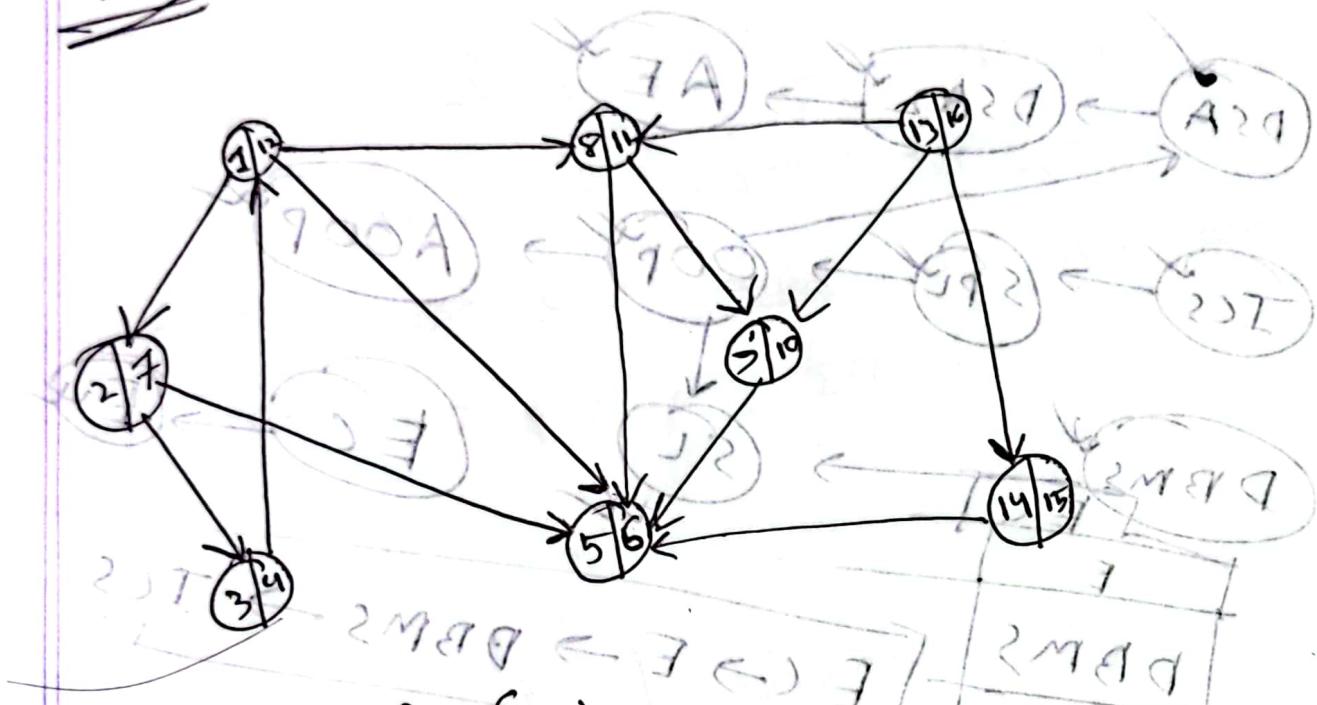


~~First enqueue is counting.~~

BFS / DFS

TC, O($V+E$)

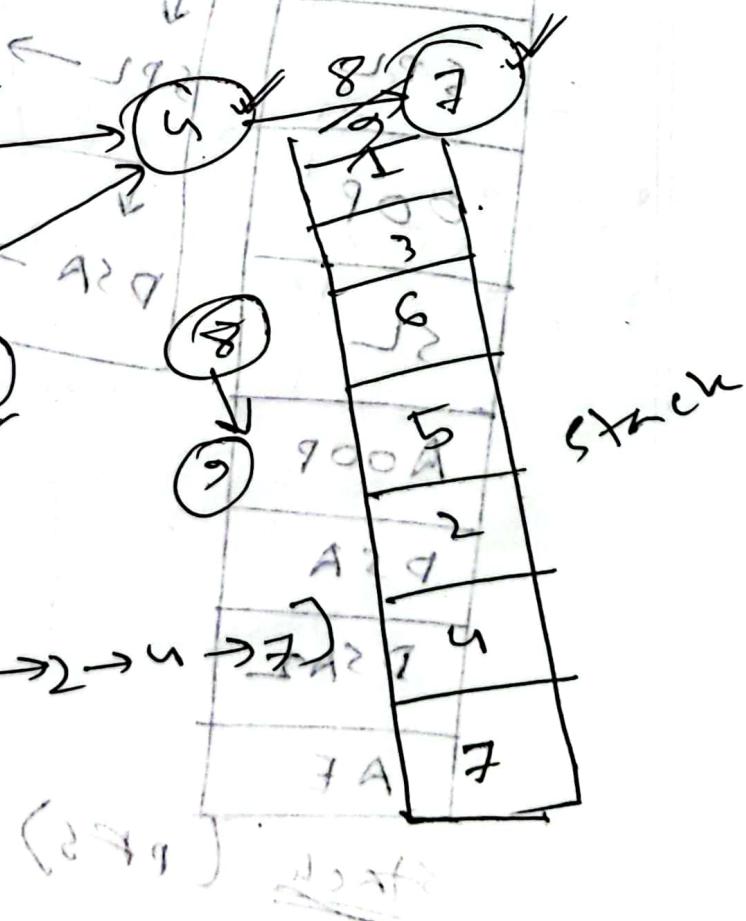
DFS



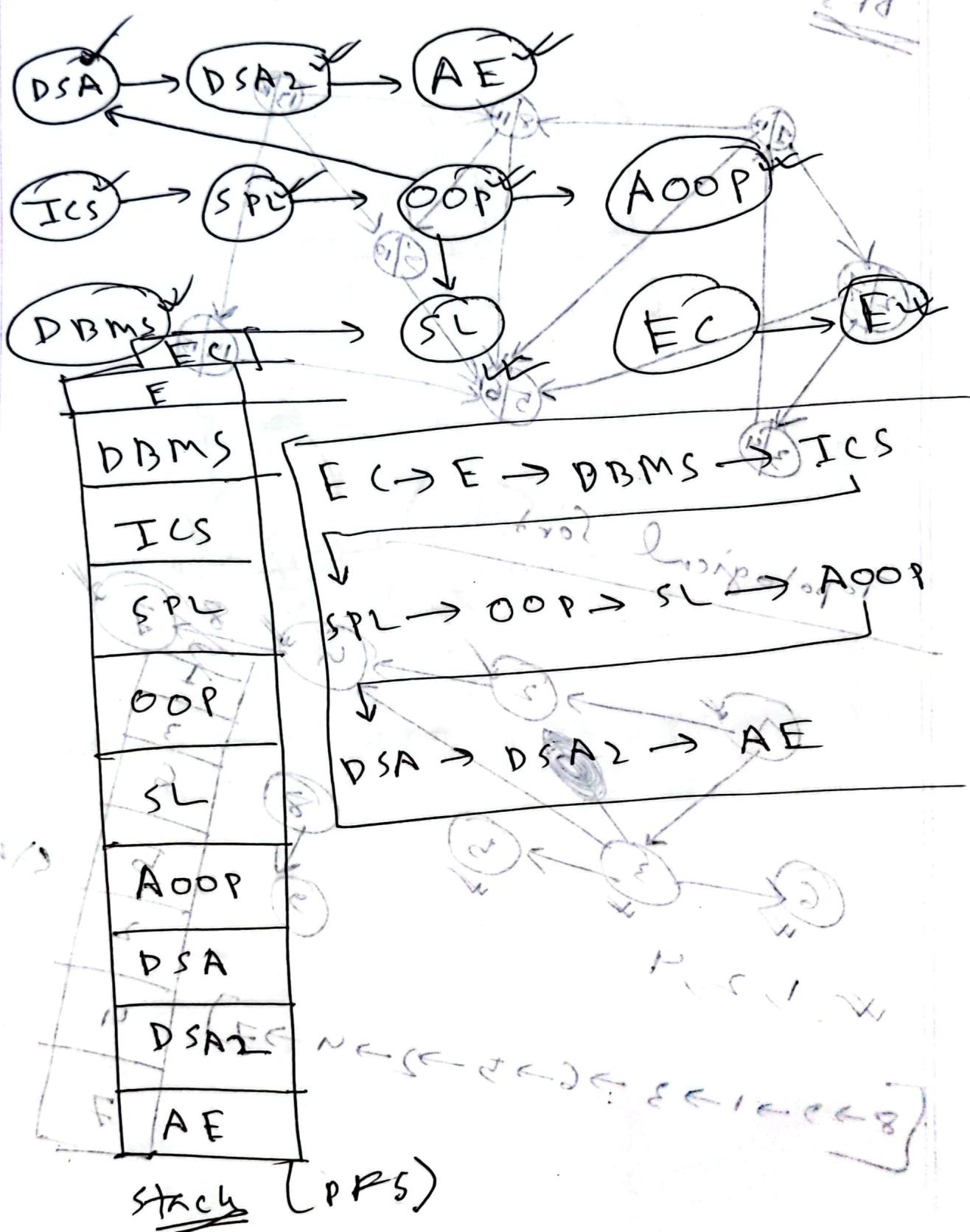
Topological Sort

w. 1, 2, 4

{ 8 → 1 → 3 → 6 → 5 → 2 → 11 → 7 }



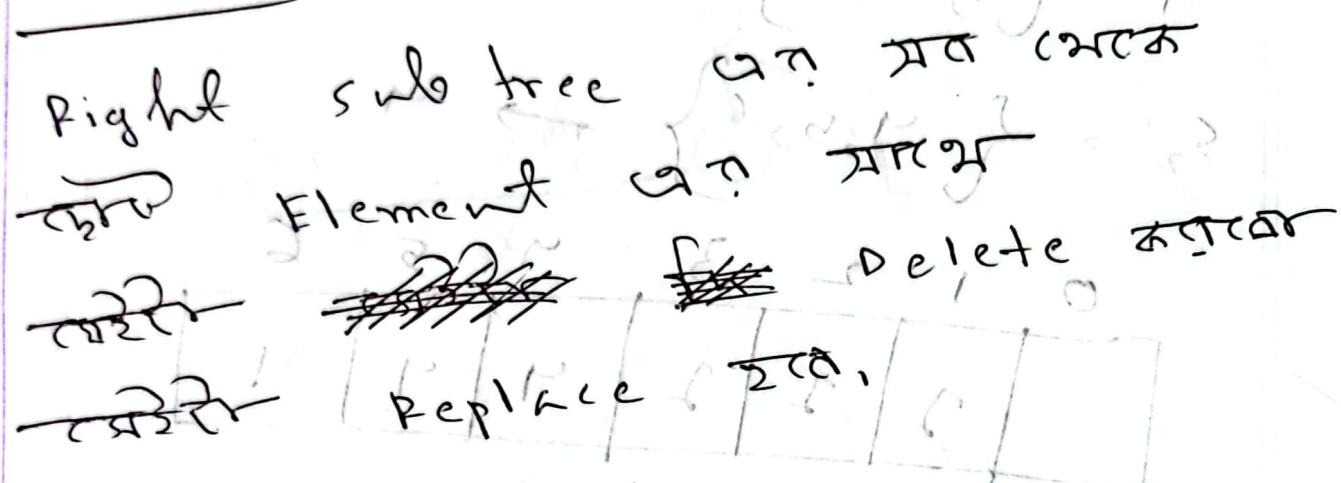
Topological sort



Lecture

3) binary tree

delete



Set

$$S_1 = \{1, 2, 3\}$$

$$S_2 = \{4, 5, 6\}$$

find(3)

union(3, 2)

\uparrow
 S_1 S_2

Union (3, 6)
 \uparrow \uparrow
 S_1 S_2

$$S_3 = S_1 \cup S_2$$

$$= \{1, 2, 3, 4, 5, 6\}$$

~~17~~

~~6W
26.8.23~~

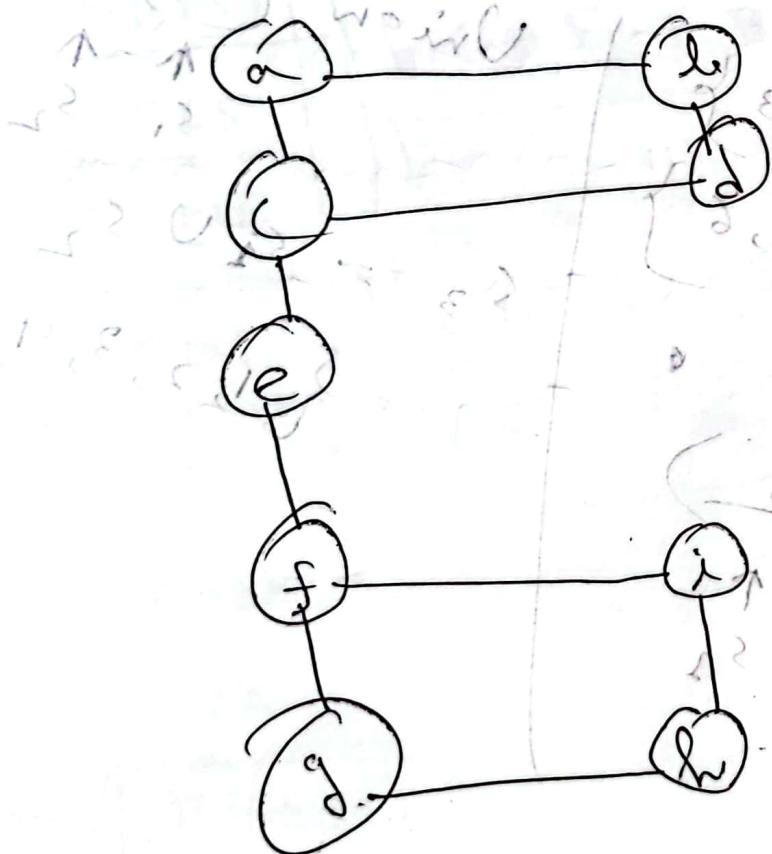
disjoint set

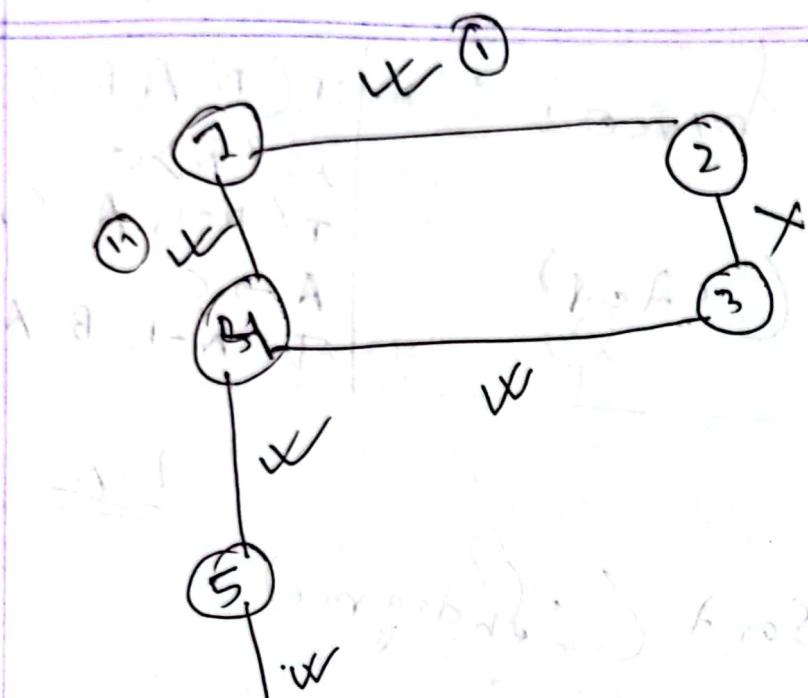
DSA 1

$$S_1 = \{1, 2, 3\} \rightarrow 1$$

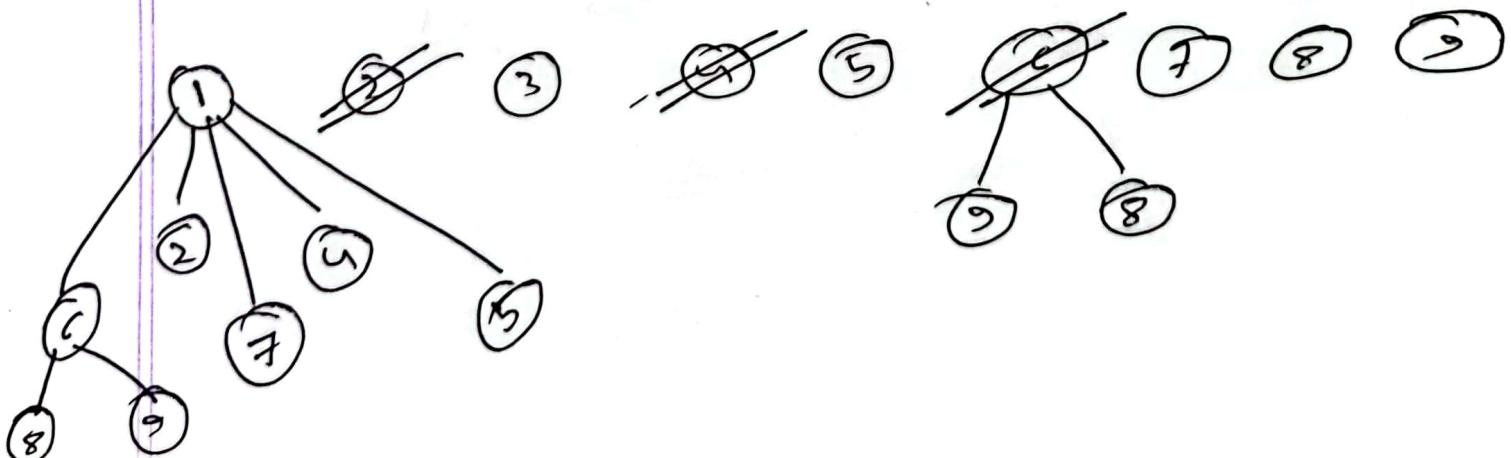
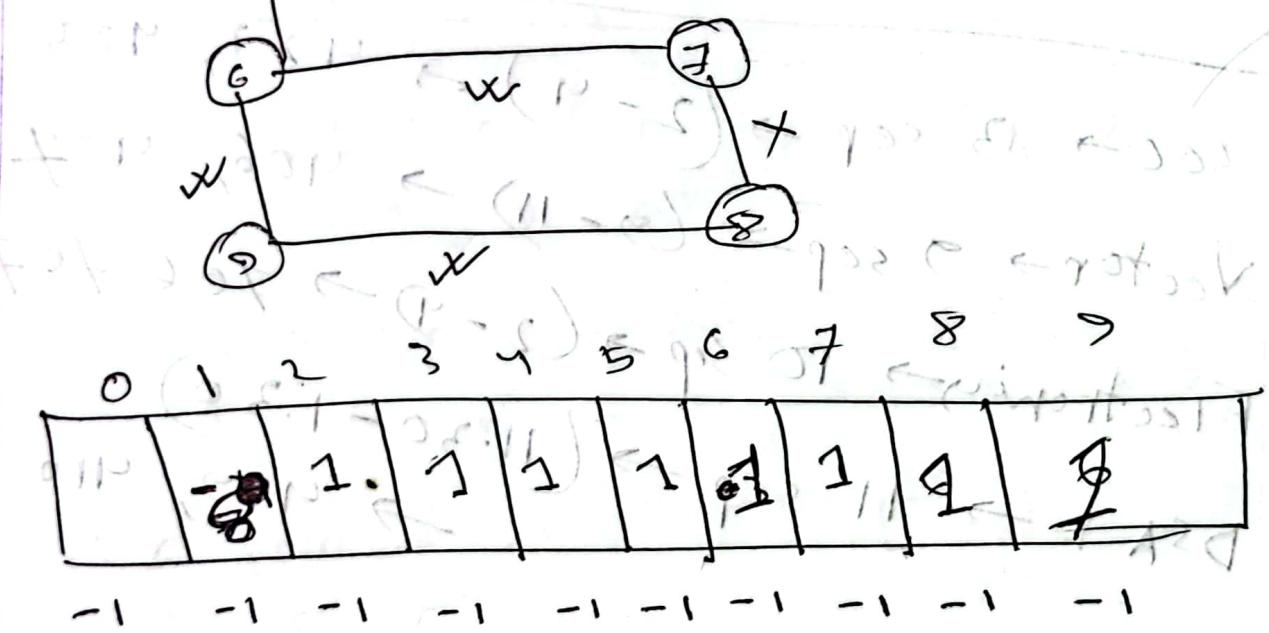
$$S_2 = \{4, 5, 6\} \rightarrow 2$$

0	1	2	3	4	5	6
2	1	2	4	4	4	4





$E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 5 \rangle, \langle 5, 6 \rangle, \langle 6, 7 \rangle, \langle 7, 8 \rangle, \langle 8, 9 \rangle, \langle 9, 10 \rangle, \langle 10, 11 \rangle \}$



heaps, BFS, DFS, TS
C.T → Tuesday

tower of hanoi : \Rightarrow

$\star 2^w - 1$ (step)

$T(N, A, B, C)$
$\text{if } (N > 0)$
$T(N-1, A, C, B)$
$A \rightarrow C$
$T(N-1, B, A, C)$

~~CW~~
~~29.8.23~~

DSA

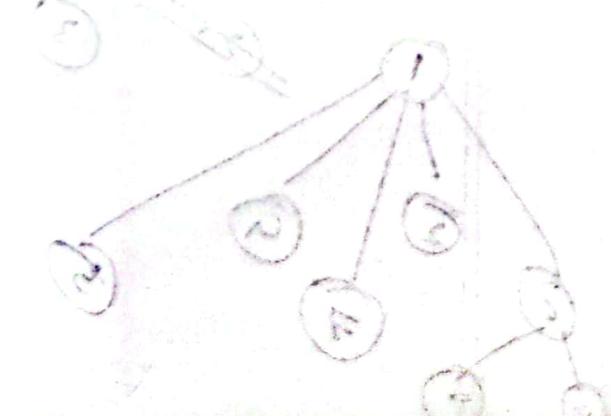
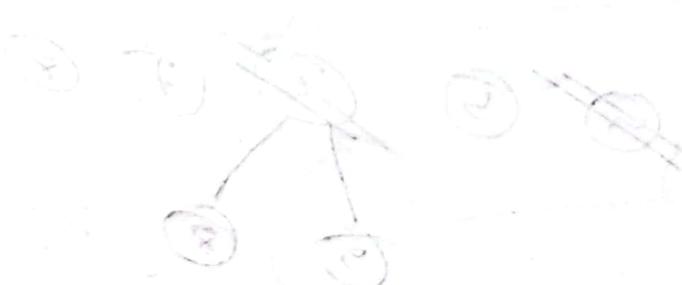
Topological Sort (Indegree)

SOC \rightarrow 3 sep $\rightarrow (2-4) \rightarrow 42^3, 425$

Vector \rightarrow 9 sep $\rightarrow (9-11) \rightarrow 406, 407$

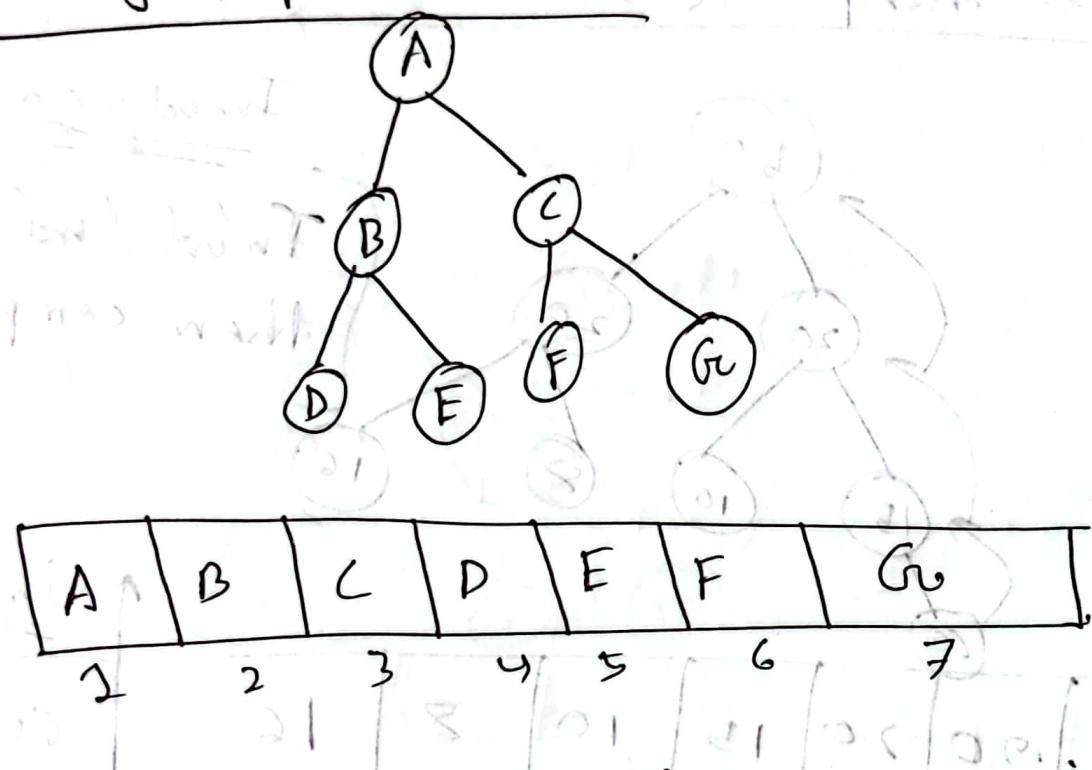
Electronics \rightarrow 10 sep $\rightarrow (2-4) \rightarrow 706, 707$

DSA 1 \rightarrow 11 sep $\rightarrow (11-30-130) \rightarrow 402, 410$

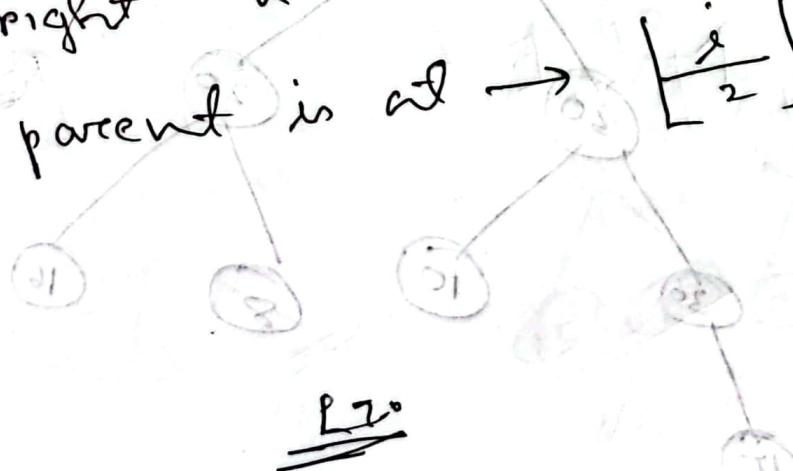


Heap

Array Representation

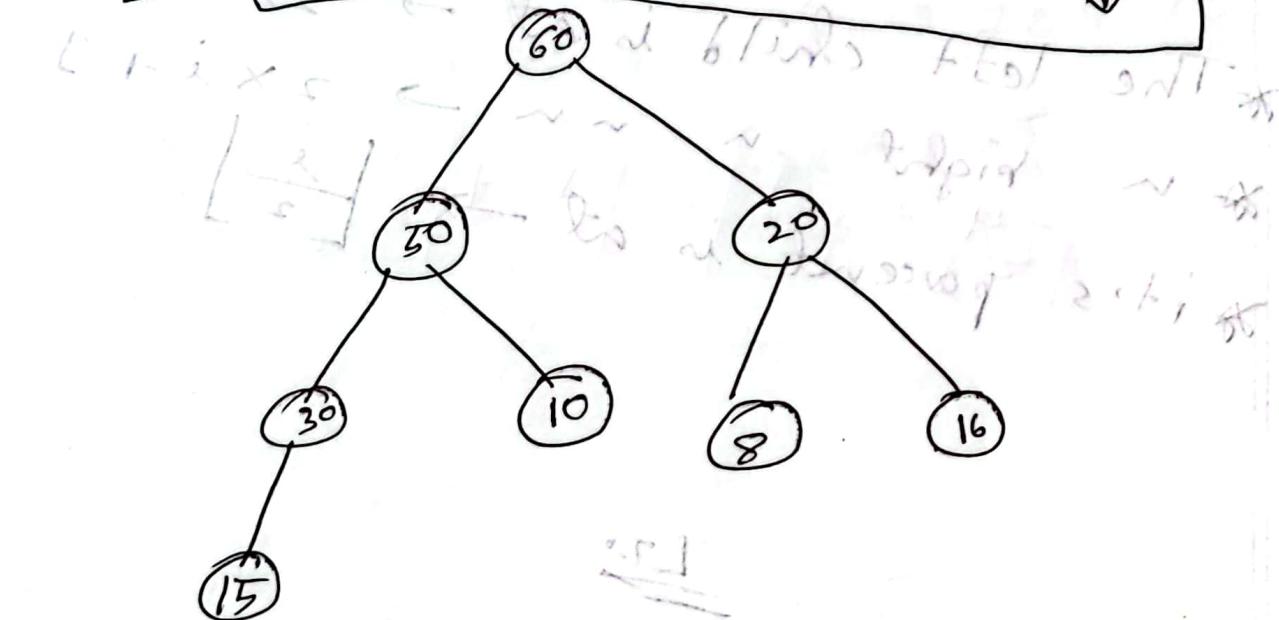
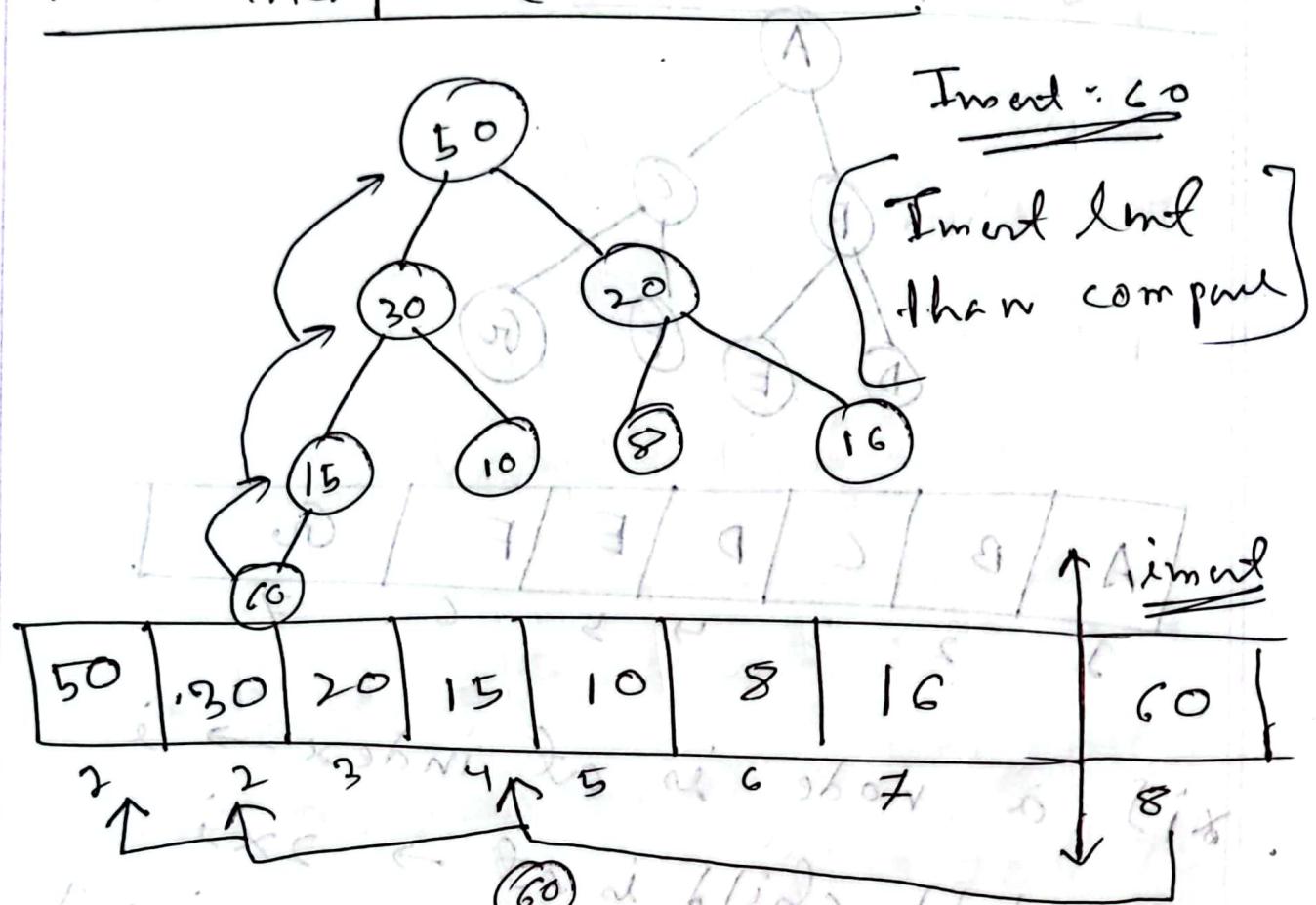


- * if a node is at index $\rightarrow i$
- * The left child is at $\rightarrow 2 \times i$
- * n right $\rightarrow 2 \times i + 1$
- * its parent is at $\rightarrow \left\lfloor \frac{i}{2} \right\rfloor$



31	101	89	101	105	105	101
8	7	5	3	2	1	0

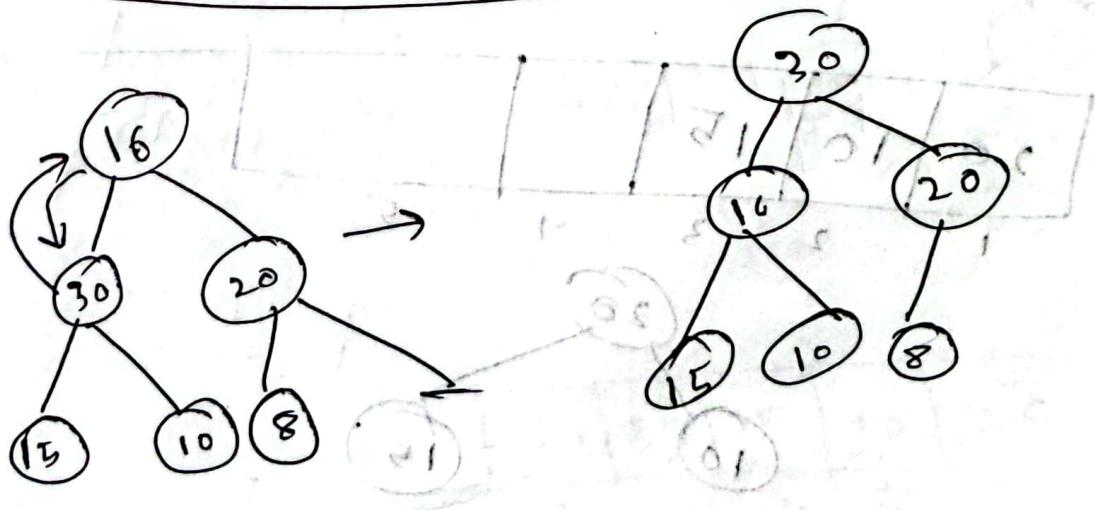
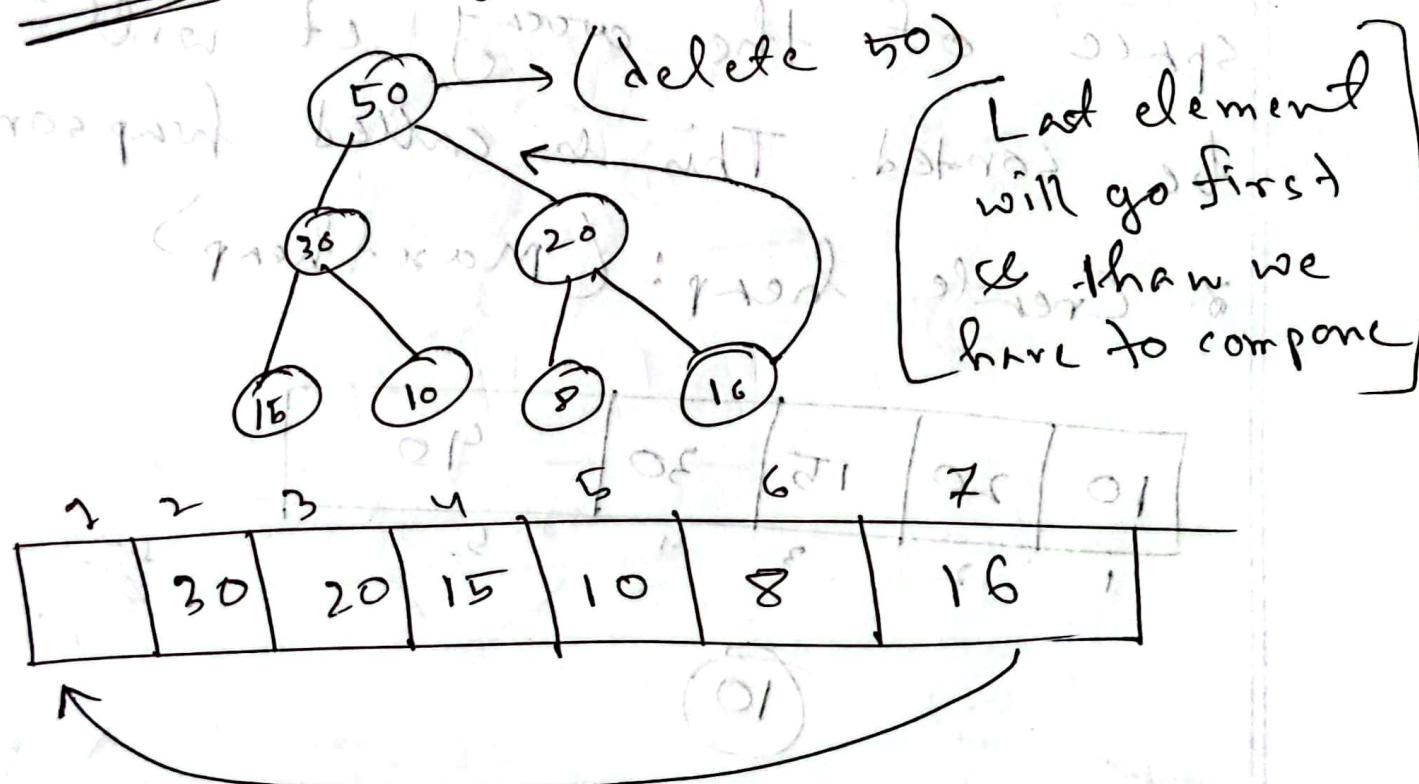
Max heap (insertion)



60	50	20	30	10	8	16	15
2	2	3	4	5	6	7	8

Time Complexity : $O(1)$ to $O(\log n)$

Delete: (only root can be deleted)



30	16	20	15	10	8
1	2	3	4	5	6

In heap sort if we delete the root & keep it in the empty space of the array it will be sorted. This is called heap sort.

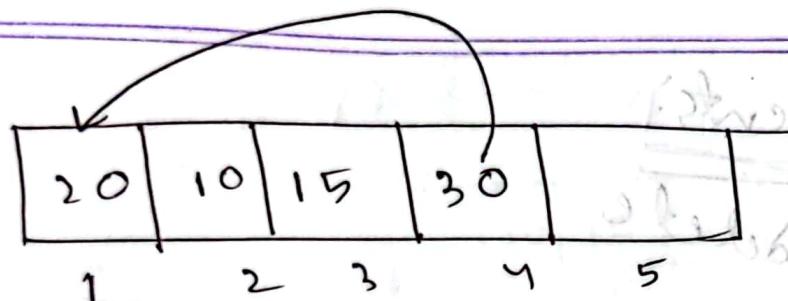
* Create heap: (Max-heap)

10	20	15	30	90
1	2	3	4	5

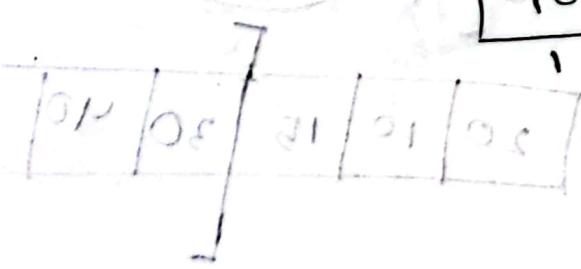
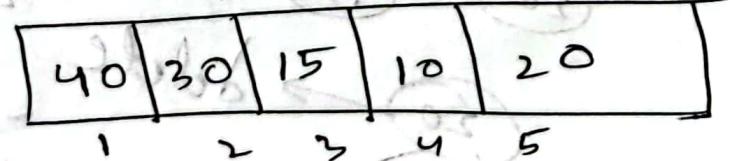
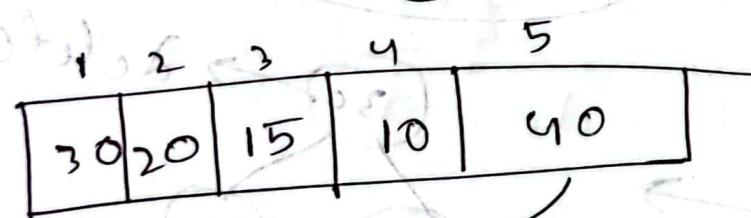
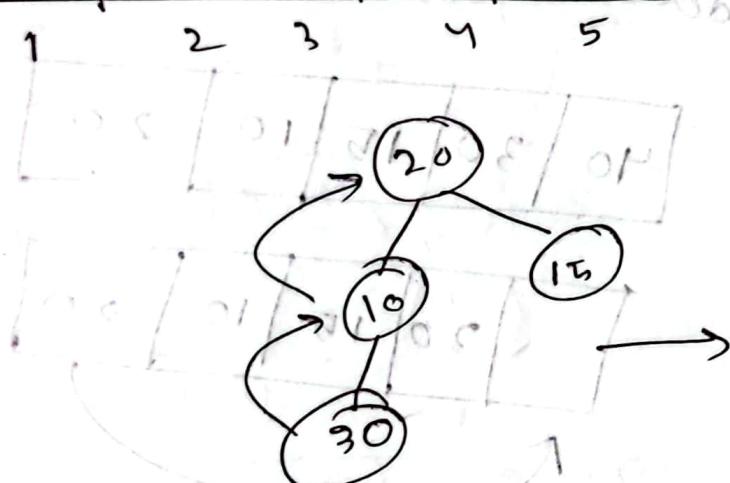
(10)

20	10	15		
1	2	3	4	5



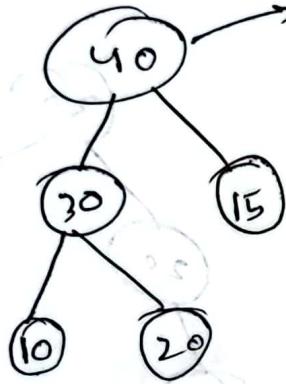


External storage



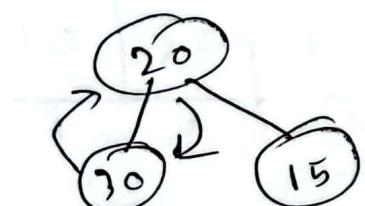
Delete Elements:

delete

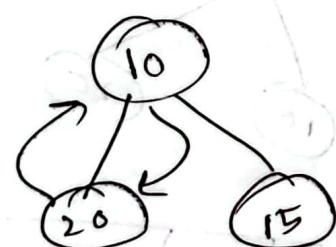
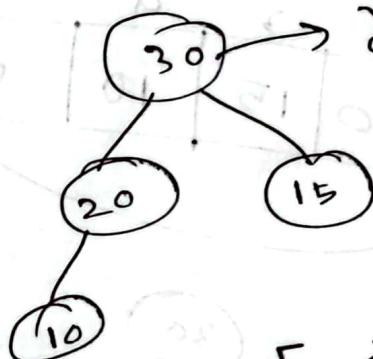


40	30	15	10	20
----	----	----	----	----

	30	15	10	20
--	----	----	----	----



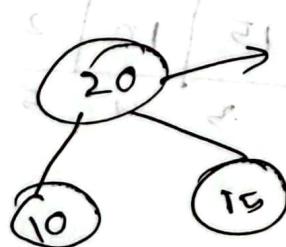
delete



30	20	15	10	40
----	----	----	----	----

Put the deleted Element in free space

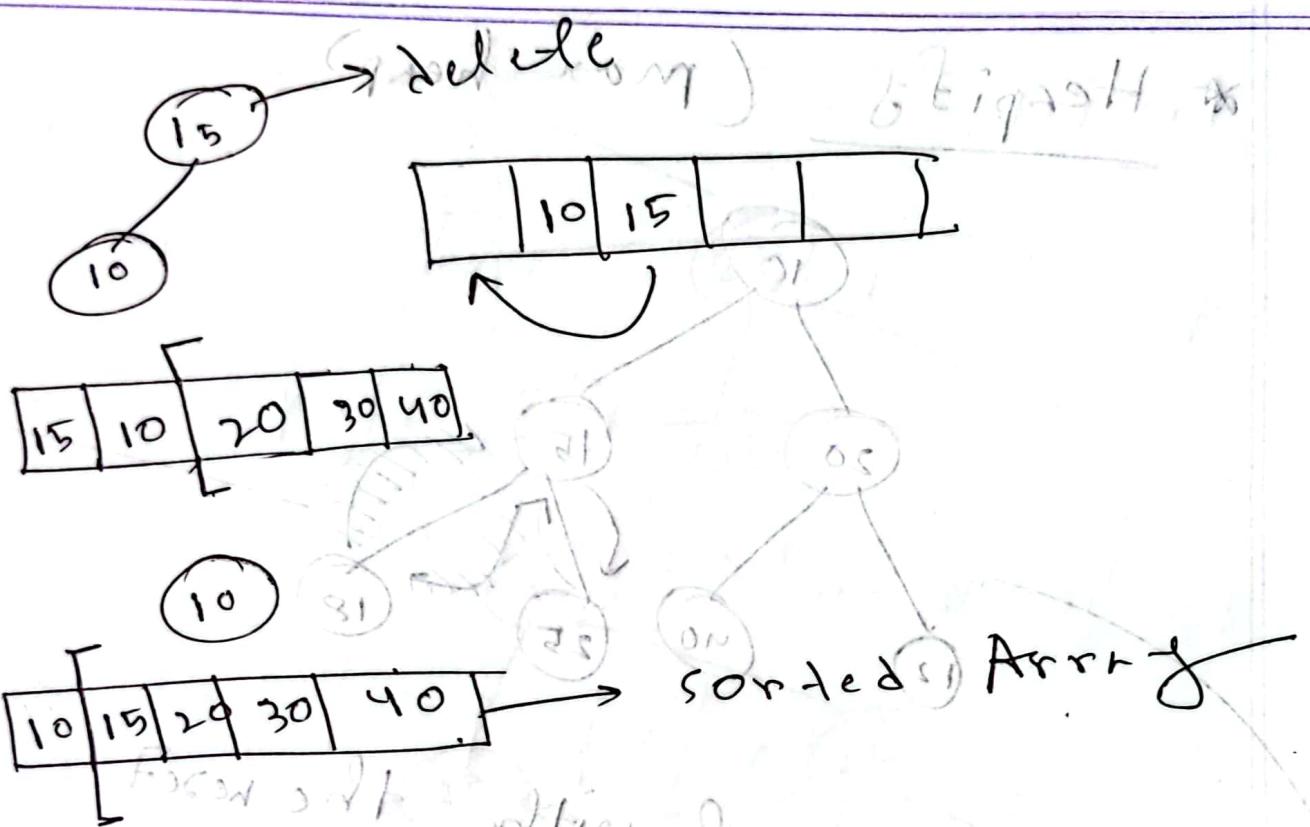
	20	15	10	
--	----	----	----	--



20	10	15	30	40
----	----	----	----	----

delete

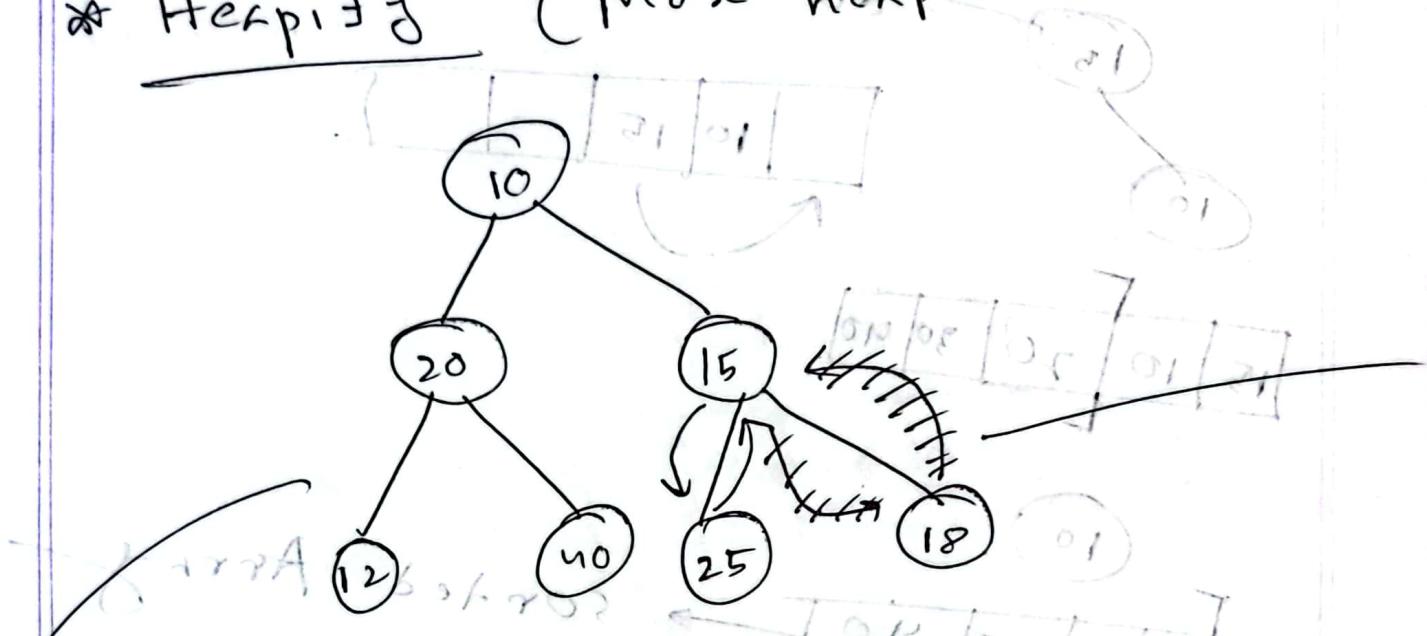




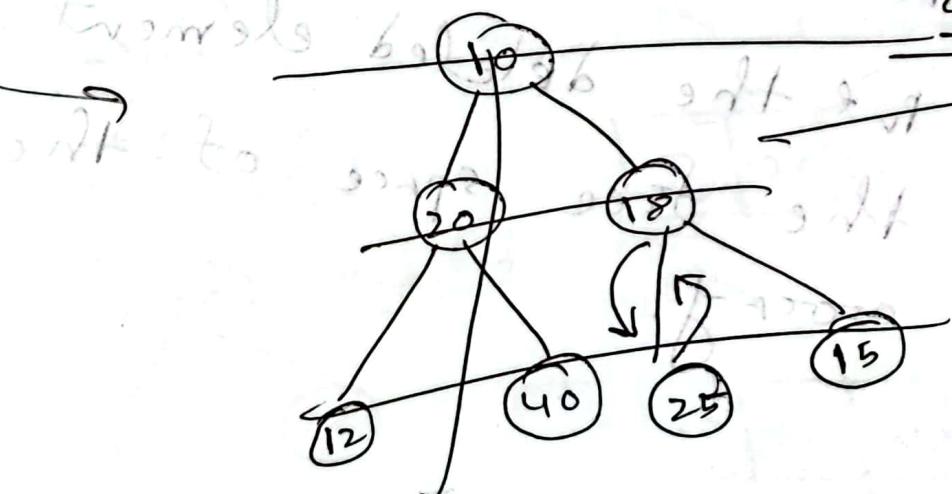
Heapsort (using BT . msubtids)

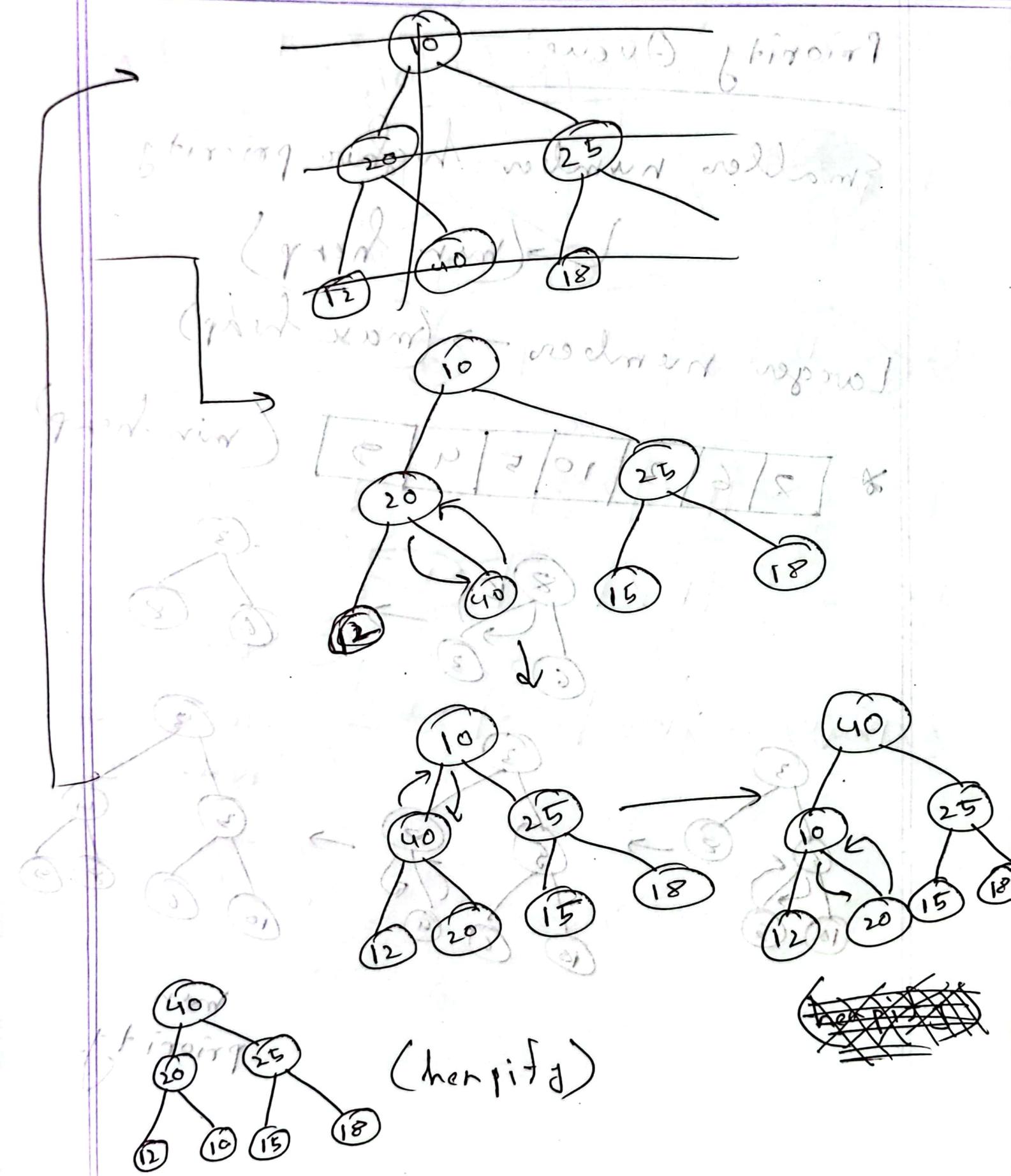
- Create heap with and in
- Delete from root
- Put the deleted element in the free space of the array.

* Heaps (max heap)



Compare parent with the next children. If parent element is less than children, swap. Start from the last. (Compare with one element)





Priority Queue:

Smaller number → higher priority

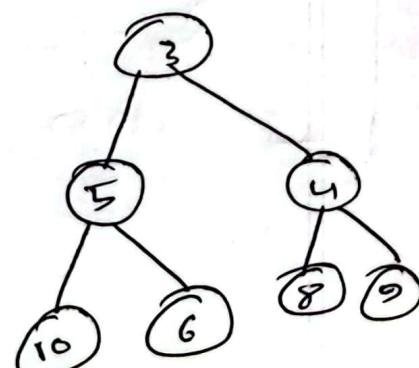
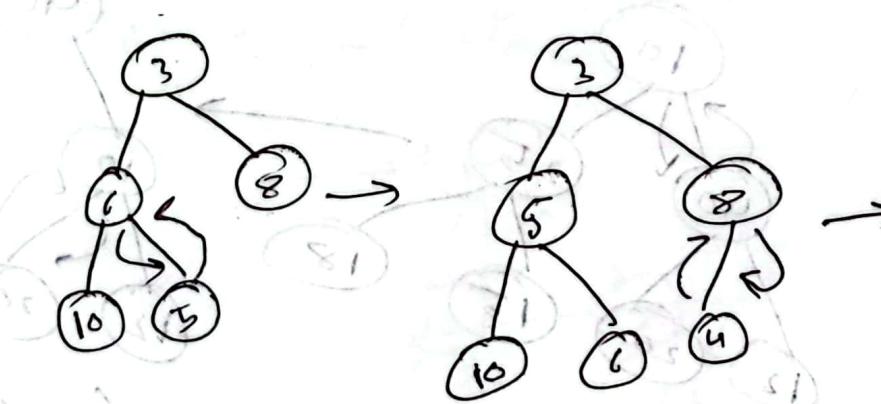
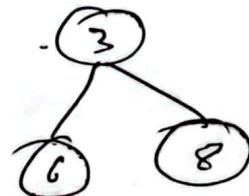
→ (min heap)

Larger number → (max heap)

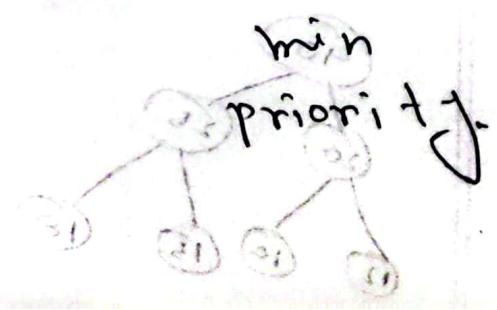
* 

8	6	3	10	5	4	9
---	---	---	----	---	---	---

(min heap)



(Optimal)



F.S.E

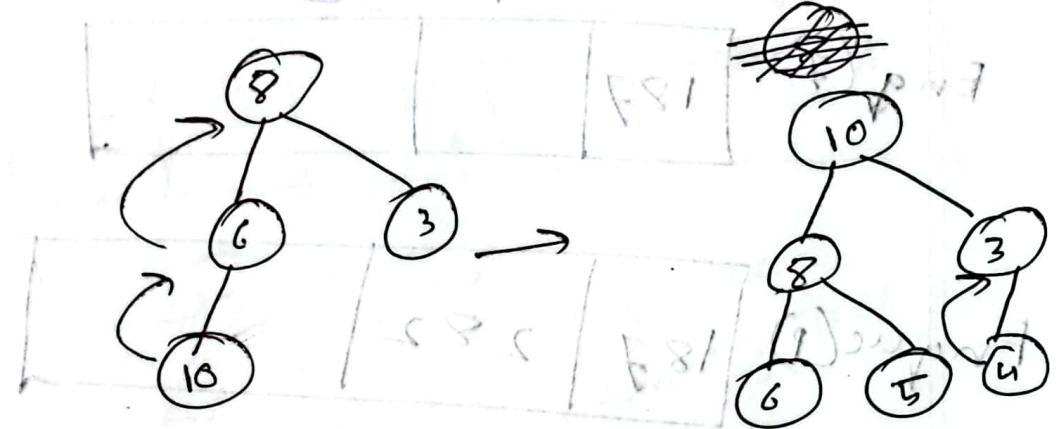
8	6	3	10	5	4	9
						(max)

Q = best

Q = min

Q = worst

Q = max

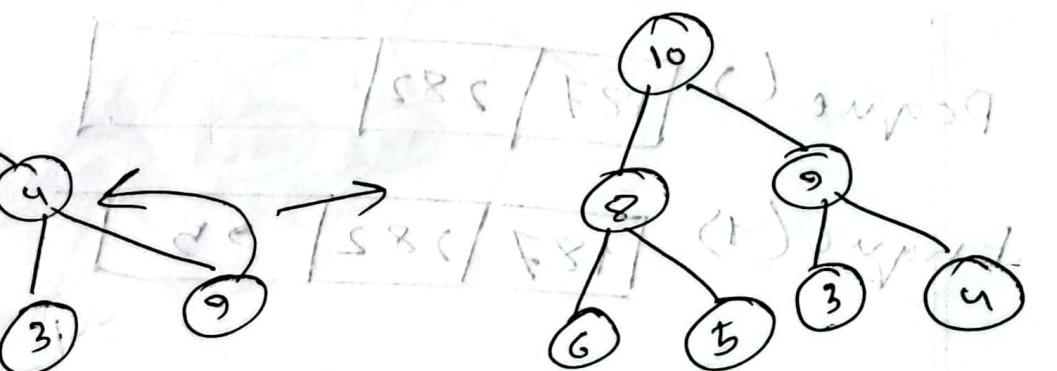


Q = best

Q = min

Q = worst

Q = max



Q = best

Q = min

max priority

~~Full~~ Sum - 27

~~2(1)~~

$$x = 92, j = 95, z = 187, p = 282$$

0 1 2

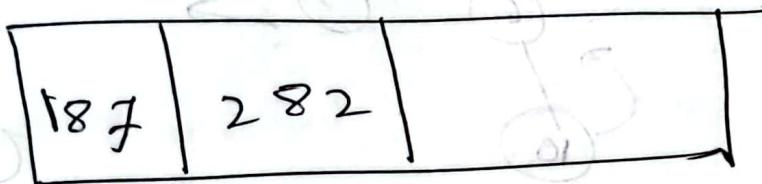
Enq(z)



front = 0

rear = 0

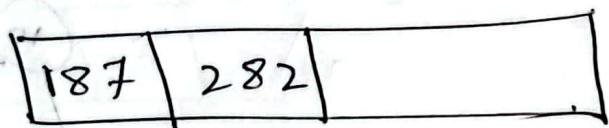
Enque(i)



front = 0

rear = 1

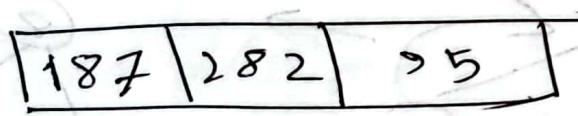
Dequeue()



front = 1

rear = 1

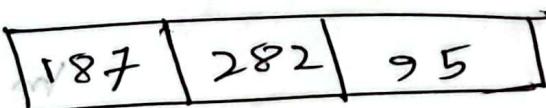
Enque(i)



front = 1

rear = 2

Dequeue()



front = 2

rear = 2

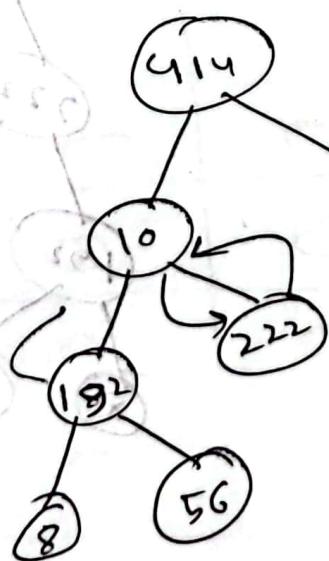
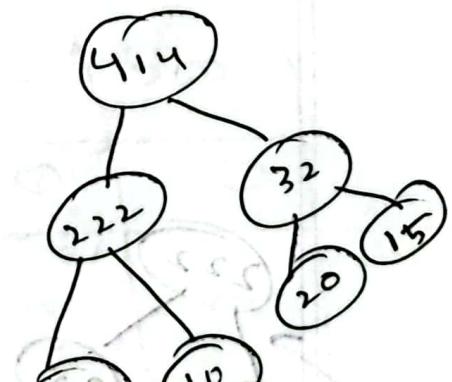
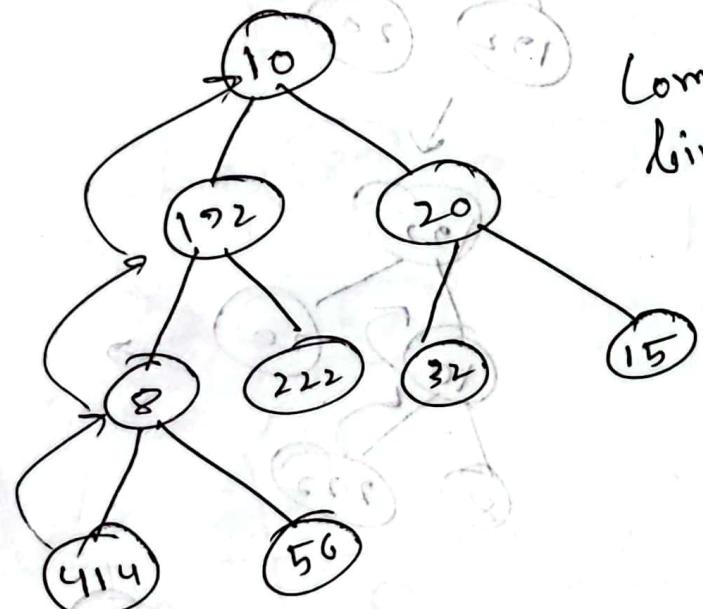
(b)

$$x = 192, \quad d = 222, \quad I = 414$$

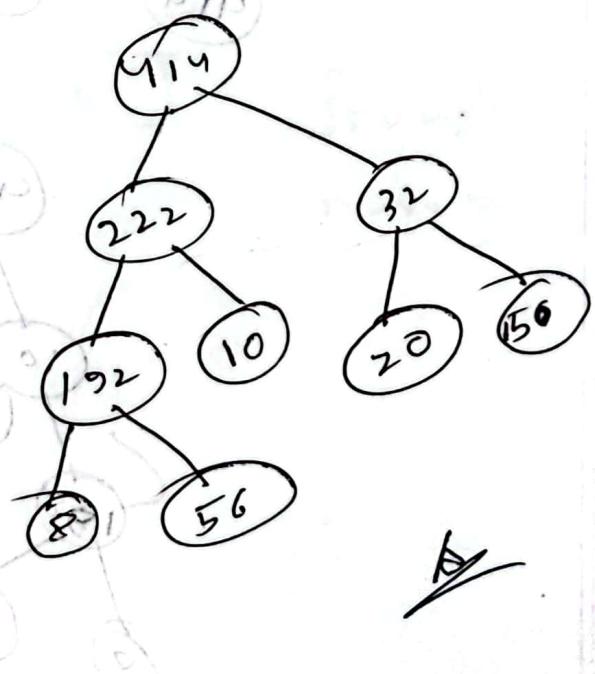
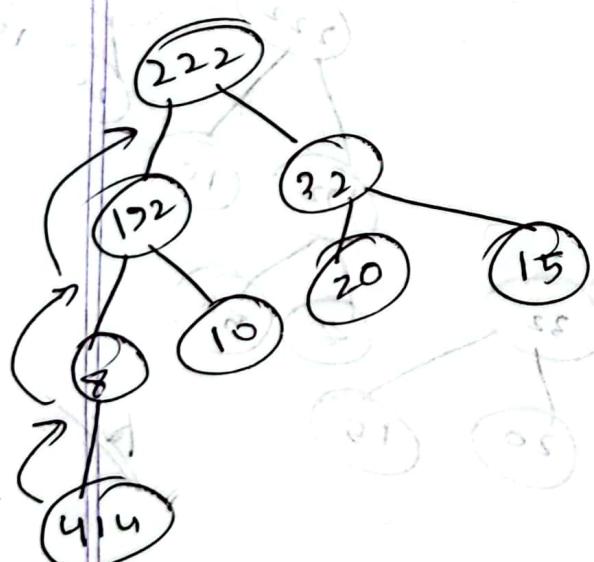
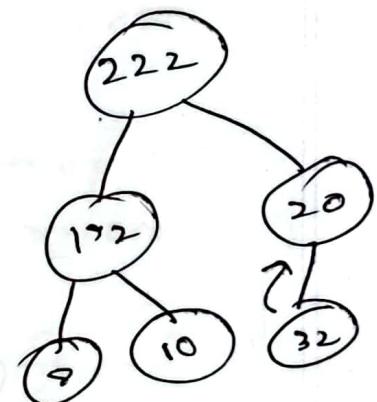
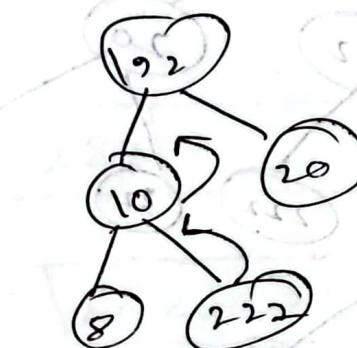
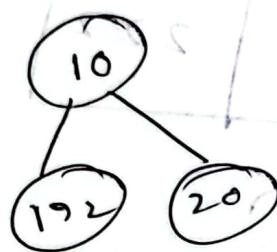
1	2	3	4	5	6	7	8	
10	192	20	8	222	32	15	414	56

Complete
binary tree

max heap



10	172	20	8	222	32	15	414	56
----	-----	----	---	-----	----	----	-----	----

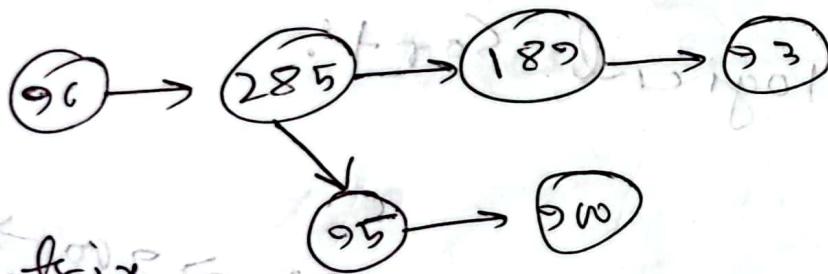


~~3p (r)~~

$x = 93, \quad J = 96, \quad Z = 189, \quad P = 285, \quad Y = 95,$
 $U = 900$

96	285	189	93	95	900
----	-----	-----	----	----	-----

(a)



(b)

Matrix

grph

96	285	189	93	95	900
96	0	1	0	0	0
285	0	0	1	0	1
189	0	0	0	1	0
93	0	0	0	0	0
95	0	0	0	0	0
900	0	0	0	0	0

list

96 → 285

285 → 189, 95

189 → 93

93 →

95 → 900

900 →

(c) topological sort:



stack

96 → 285 → 95 → 900 → 189

↓

93

b

class

method

return

else

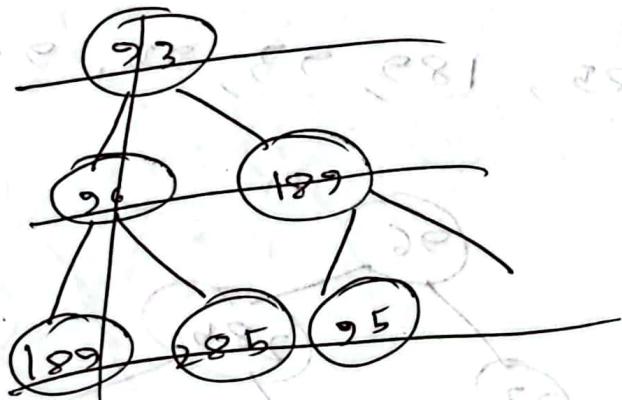
if

for

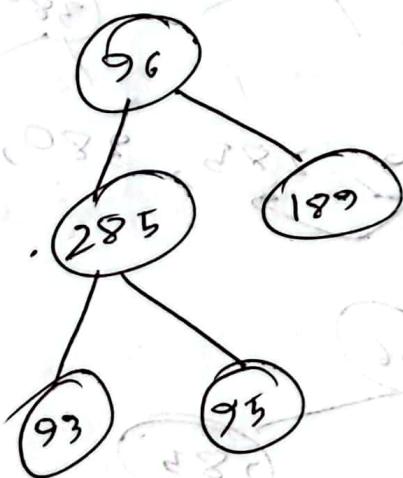
while

u(α)

$\alpha = 93 + 285 + 189 + 285 + 95$
 $x = 93, \beta = 96, \gamma = 189, \rho = 285, \tau = 95$



$\beta, \rho, \gamma, x, \tau$



BFS: $96 \rightarrow 285 \rightarrow 189 \rightarrow 93 \rightarrow 95$

P.T.O

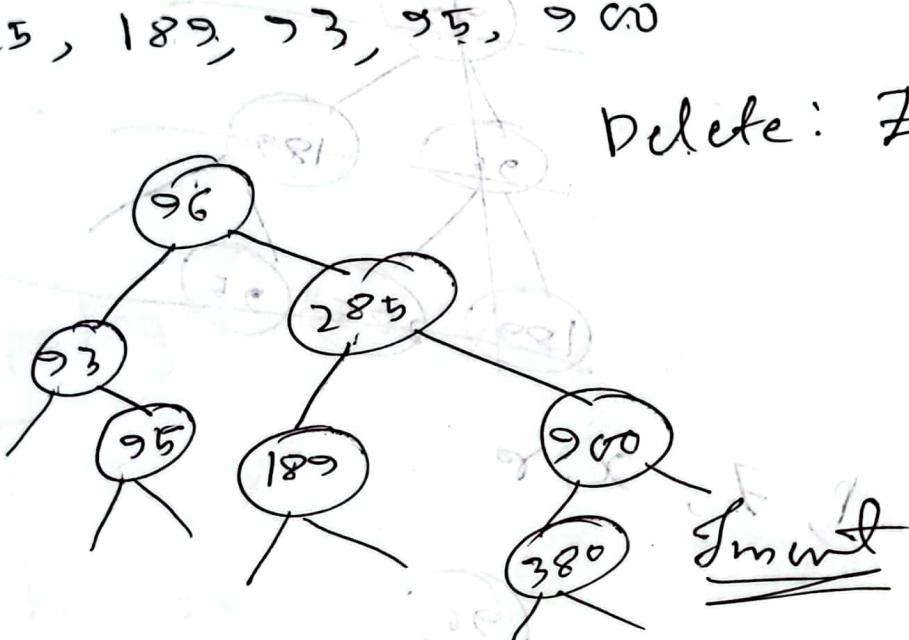
~~4 (b)~~

$x = 73, J = 76, Z = 189, P = 285, S = 95,$

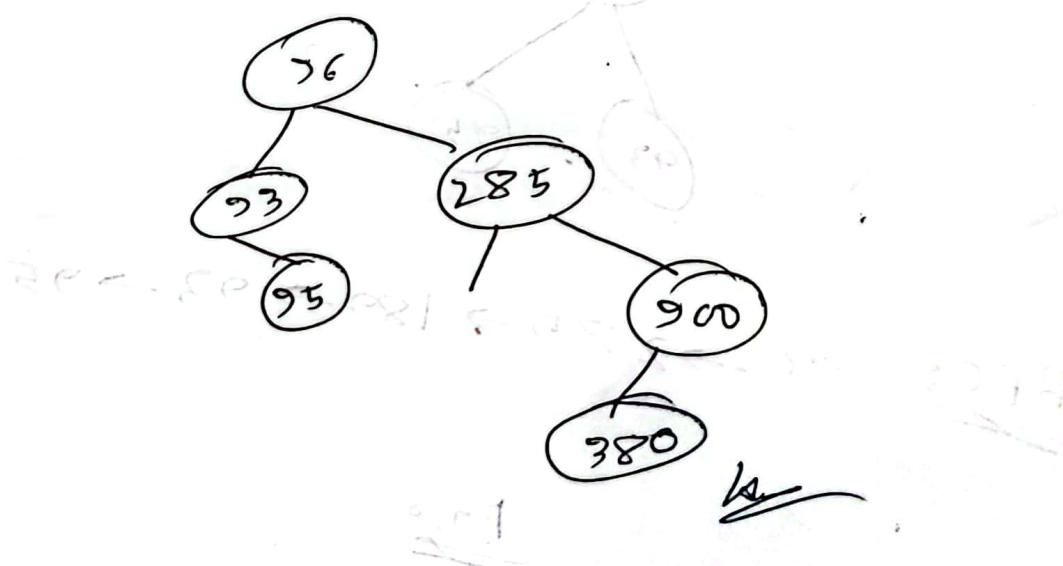
$P + R = 380$ (~~Sum~~) $t = 900$

$76, 285, 189, 73, 95, 900$

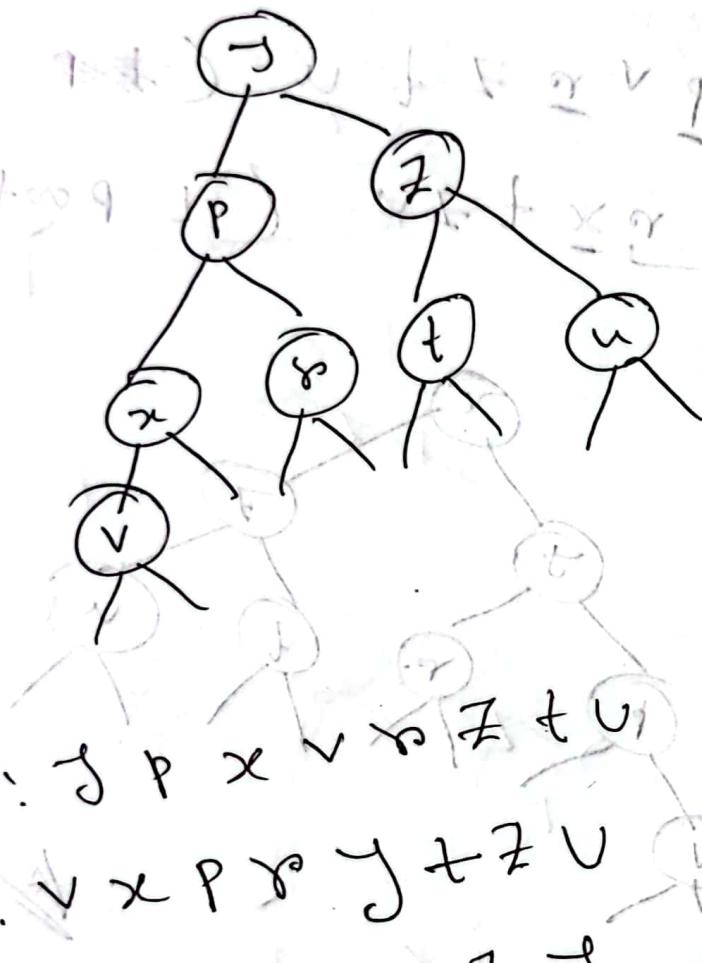
Delete: Z



$73, 95, 76, \cancel{189}, 285, 380, 900$



~~(a)~~



~~(b)~~

pre: J P x v w Z t u

in: v x P w J + Z u

post: v x w P t u Z J

Level: J P Z x w t u v

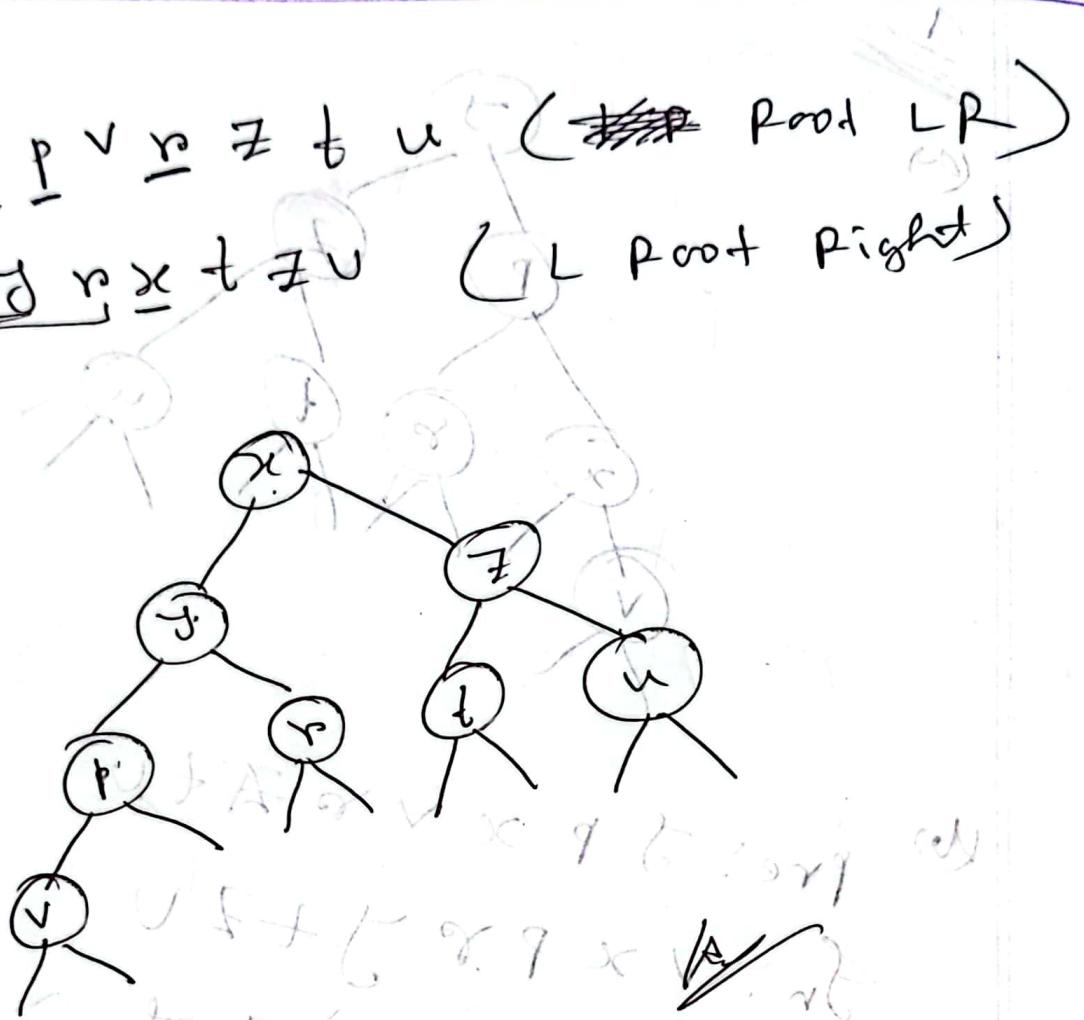
height: 3

~~key~~

(c)

Pre: x j p v r z t u (~~Root~~ Root LR)

In: v p j r x t z u (L Root Right)

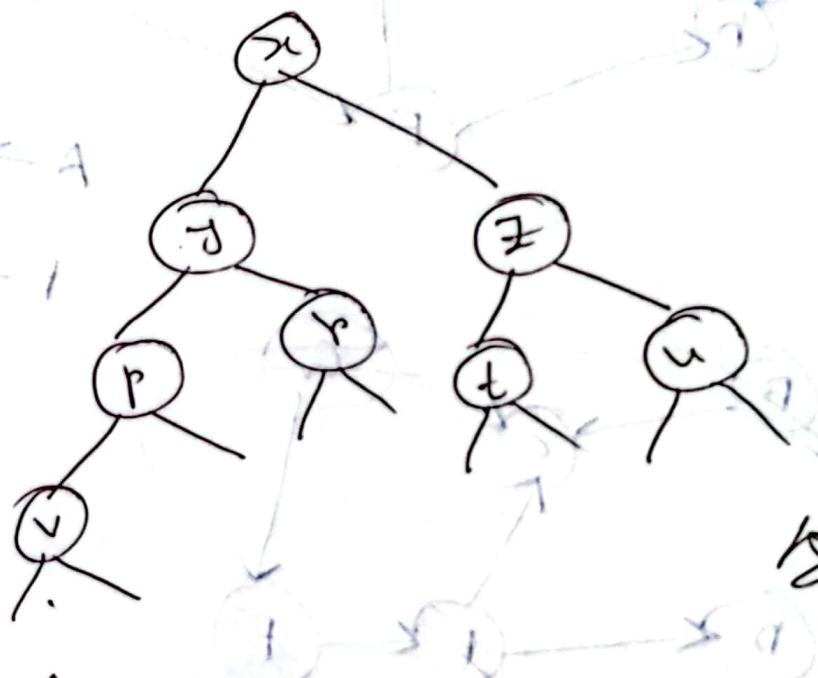


v < p < r < t < l < u

x < j < p < r < v < t < l < u

Fall-21

(i) In: $v \underline{p} \underline{j} \underline{r} \underline{x} \underline{t} \underline{z} \underline{v}$ [L Root R.]
Post: $v \underline{p} \underline{r} \underline{j} \underline{t} \underline{u} \underline{z} \underline{x}$ [L R Root]

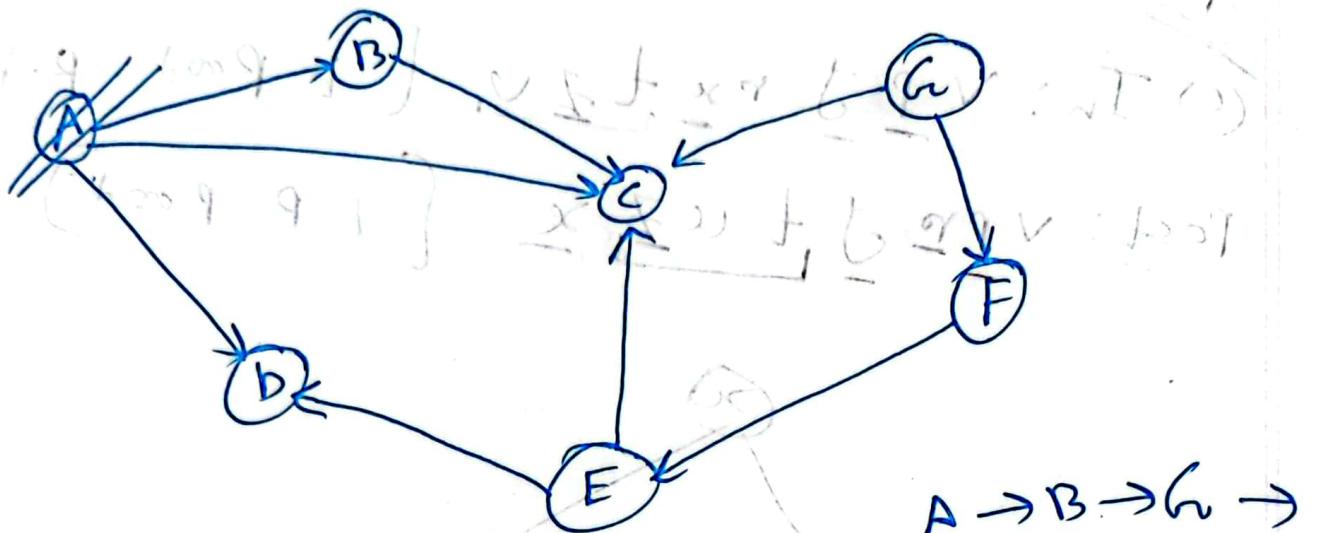


(ii) Level Order

Q	*	*	*	*	*	*	*	*
---	---	---	---	---	---	---	---	---

DFS: $x \rightarrow j \rightarrow z \rightarrow p \rightarrow r \rightarrow t \rightarrow u \rightarrow v$

BFS:



$A \rightarrow B \rightarrow C \rightarrow$

$F \rightarrow E \rightarrow C \rightarrow D$

