Mohammad Ali Jinnah University

Chartered by Government of Sindh - Recognized by HEC

FINAL ASSIGNMENT

Name: Muhamad Fahad

Id: FA19-BSSE-0014

Subject:

Section: AM

Teacher: Awais Ahmed

Date: Saturday, January 23, 2021

FINAL ASSIGNMENT

Dear Students,

Take your own two examples (input size of n >=8) for each of the topics we covered and show proper working (dry run).

Take pictures and make a single pdf file

1.) Asymptotic Notation

Code:

```
int i;
for(int j=0;j<i;j++){
    for(k=0;k<i;k++){
        for(int l=0;l<i;l++){
            Printing i-1;
        }
        i=0;
    }
}
```

Output:

The time compelex of the altrogthm is (n (for the first loop x n (for the first inner loop) x n (for last loop))

$n \times n \times n \Rightarrow n^3$

Best case: n^3

Avg case n^3

Worst case: n^3

Example 2:

Code:

```
int i=n;
for(int a=i;a<i*i;a++){
    while(a%2==1){
        a++
    }
}
```

Output:

- first loop will run n time
- second while loop will run's n/2 time

so,

$n \times (n/2) = n^2/2 \Rightarrow n^2$

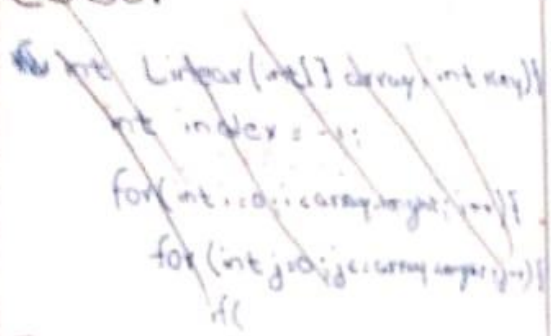
Best case: n times.

Worst Case: n^2 times.

2.) Searching:

A.) Linear Search:

Code:



Example 1:

array = {1, 8, 2, 82, 67, 32, 22, 37}

Key = 37

i	arr[i]	Key	condition
0	1	37	false
1	8	37	false
2	2	37	false
3	82	37	false
4	67	37	false
5	32	37	false
6	22	37	false
7	37	37	True.

Example (output = 2)

The given array = {7, 27, 33, 52, 1, 82, 15}

and the key = 33

i	arr[i]	Key	arr[i] != key	Condition
0	7	33	7 != 33	false
1	27	33	27 != 33	False
2	33	33	33 == 33	true.

Time Compel

Best Case: O(1)

Avg Case: O(n/2)

Wrong Case: O(n)

B.) Binary Search:

Time Complexity

Best Case: O(1)

Avg Case: O(n/2)

Wrongt Case: O(n)

Example 1:

if array {1,2,3,4,5,6,7,8}

Key = 6

Start	end	mid	arr[mid]	key con
0	7	3	4	6 fals
4	7	5	6	6 true

Example 2:

A = {37,52,72,78,79,80,81,

Key = 37

Start	end	mid	arr[mid]	key cond
0	7	3	78	37 F
0	2	1	52	37 F
0	0	0	37	37 T
0	0			

B) Binary Search:

Time Complexity

Best Case: O(1)

Avg Case: O(n/2)

Wrongst Case: O(n)

Example 1:

if array {1,2,3,4,5,6,7,8}

key = 6

Start	end	mid	arr[mid]	key	cond
0	7	3	4	6	false
4	7	5	6	6	true

Example 2:

A = {37,52,72,78,79,80,81,}

Key = 37

Start	end	mid	arr[mid]	key	cond
0	7	3	78	37	F
0	2	1	52	37	F
0	0	0	37	37	T
0	0				

SORTING:

•) Bubble Sort:

Time Complexity:

Best case: O(

Avg Case: O(

Wrost Case: O(n²)

Example 1:

A={7,77,22,52,78,5,8,1}

i	j	arr[i]	arr[j]	arr[i]<arr[j]	Condition(bool)
0	0	7	77	7<77	True
0	1	7	22	7<22	True
0	2	7	52	7<52	True
0	3	7	78	7<78	True
0	4	7	5	7<5	True
0	5	7	8	7<8	True
0	6	7	1	7<1	false
0	7	1	77		

∴ 0 8 {1,77,22,52,78,5,8,7}

7	0	78	1	78<1	false
7	1	78	5	78<5	false
7	2	78	8	78<8	false
7	3	78	22	78<22	false
7	4	78	52	78<52	false
7	5	78	77	78<77	false
7	6	78	78	78<78	false
7	7				

7 7 {1,5,7,8,22,52,77,78}

2.) Insertion Sort:

- Time Complexity

Best Case:
Avg Case:
Wrost Case:

-) Example #1:~

A = {52, 77, 18, 19, 11, 44, 48, 84}

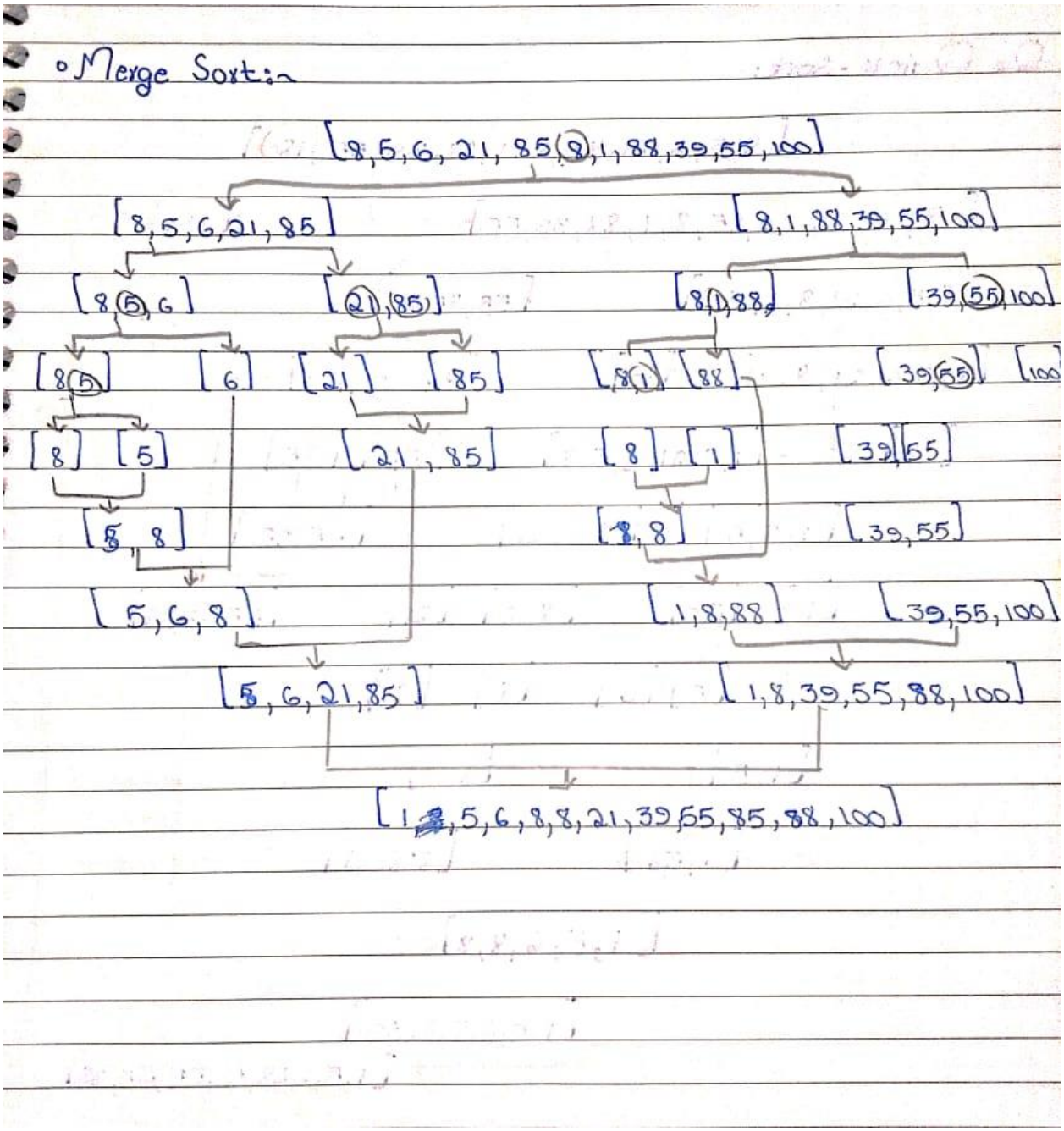
i	j	Key	arr[j]	Condition	arr
1	0	77	52	false	{52,77}
2	1	18	77	True	{52,77,77}
2	0	18	52	T	{52,52,77}
- - - - - - -			i=2		{18,52,77} - - -
3	2	19	72	T	{18,52,77,77}
3	1	19	52	T	{18,52,52,77}
3	0	19	18	T	{18,19,52,77}
- - - - - - -			i=3		{18,19,52,77} - -
4	3	11	77	T	{18,19,52,77,77}
4	2	11	52	T	{18,19,52,52,77}
4	1	11	19	T	{18,19,19,52,77}
4	0	11	18	T	{18,18,19,52,77}
- - - - - - -			i=4		{11,18,19,52,77} - -
5	4	44	77	T	{11,18,19,52,77}
5	3	44	52	T	{11,18,19,52,77}
- - - - - - -			i=5		{11,18,19,44,52,77} - -
6	5	48	77	T	{11,18,19,44,48,52,77}
6	4	48	52	T	f "
6	3	48	44	F	f "
- - - - - - -			i=6		{11,
- - - - - - -			i=7		{11,18,19,52,77,78}

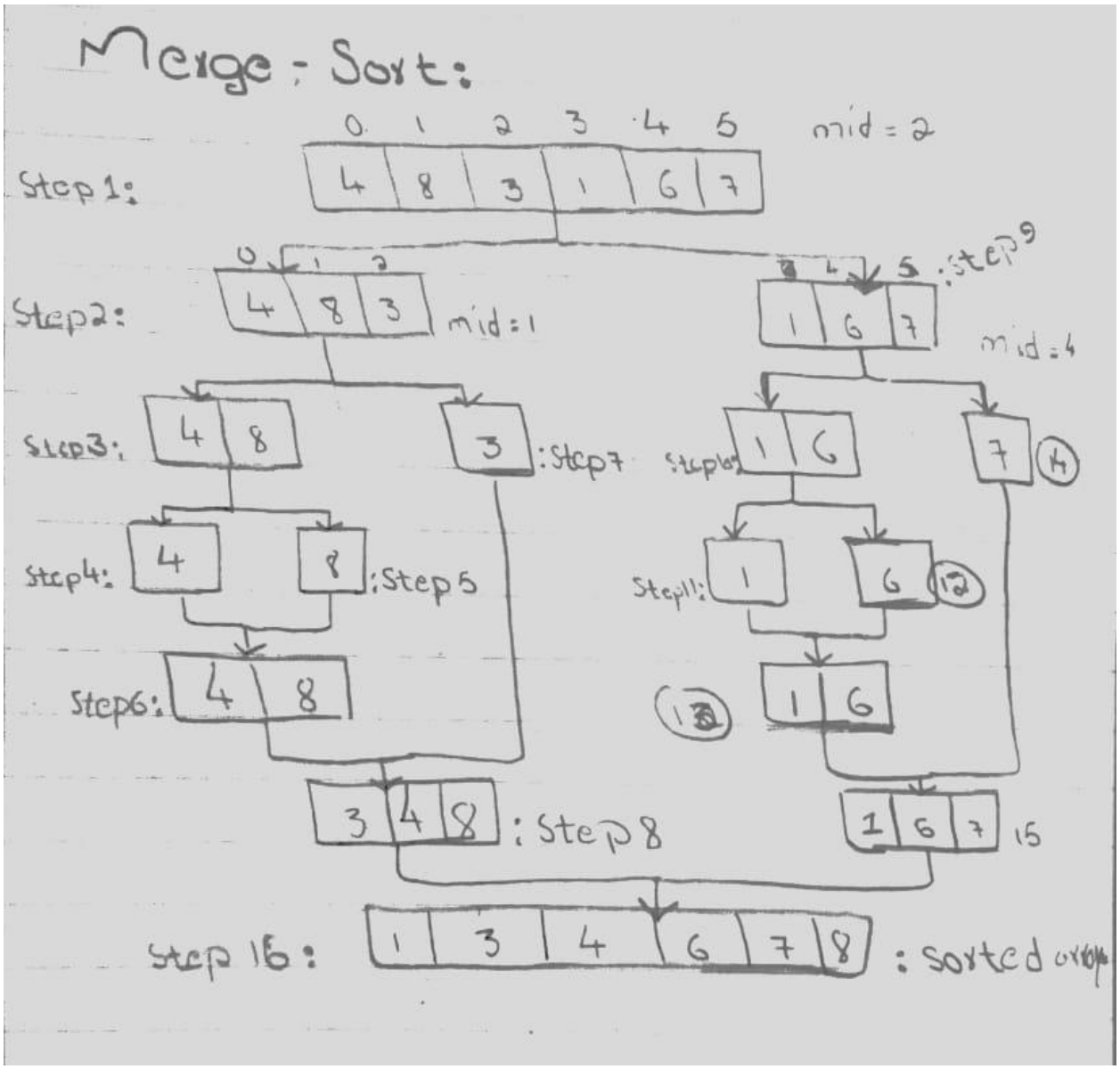
Example 2:

A = {1, 8, 32, 4, 55, 58, 78, 94}

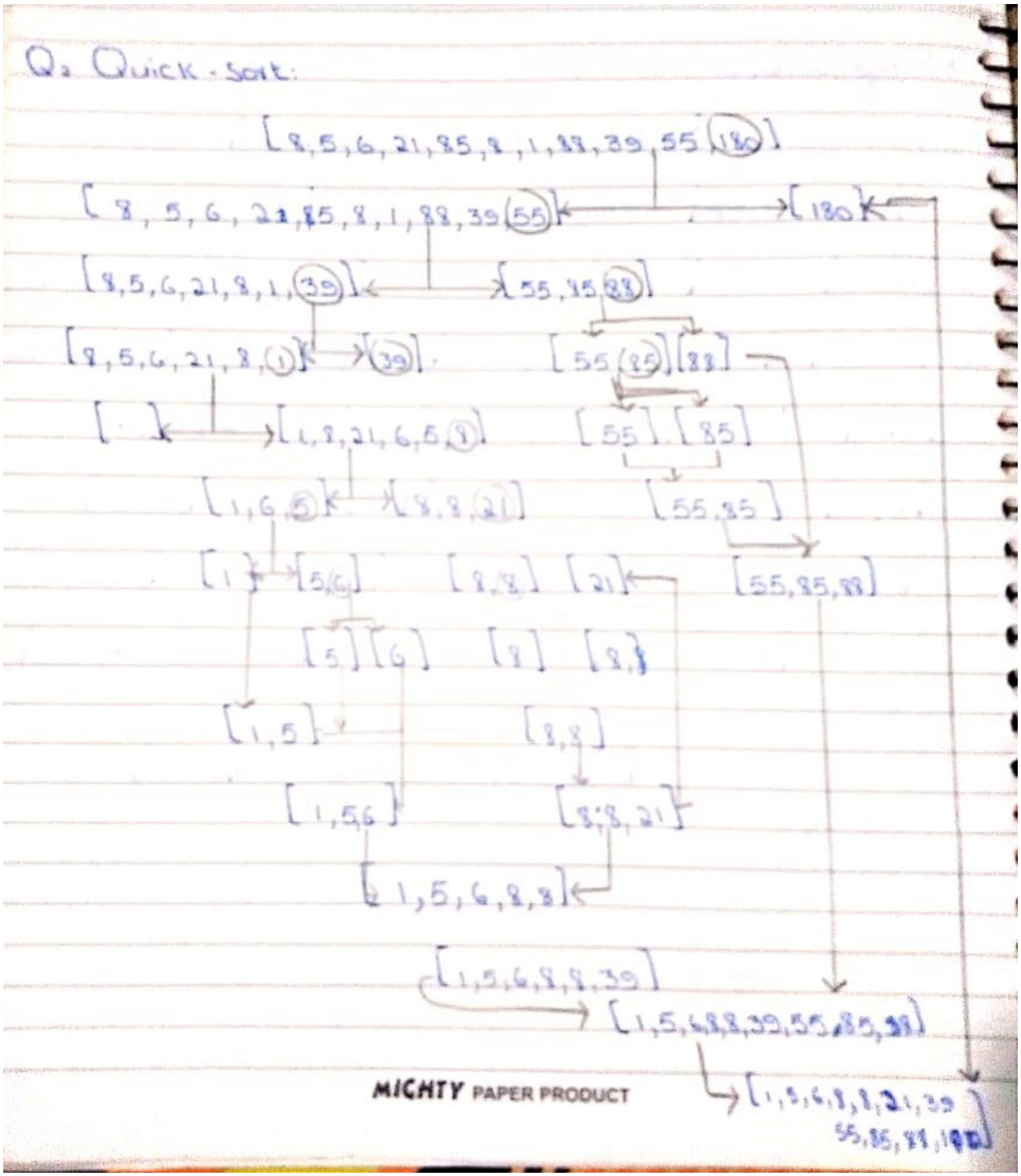
i	j	Key	arr[j]	Condition	arr
2	1	8	32	False	{1,8,32}
3	2	32	8	false	{1,8,32,32}
4	2	4	8	false	{1,8,8,32}
4	0	4	1	false	{1,8,8,32}
5	4	55	32	false	{ }
6	5	58	55	False	{ }
7	6	78	58	false	{ }
8	7	94	78	false	{ }

- Merge Sort:-


[8,5,6,21,85,8,1,88,39,55,100]
[8,5,6,21,85]
[8,1,88,39,55,100]
[8,5,6]
[21,85]
[8,1,88]
[39,55,100]
[8,5]
[6]
[21]
[85]
[8,1]
[88]
[39,55]
[100]
[8]
[5]
[21,85]
[8]
[1]
[39]
[55]
[5,8]
[8,8]
[39,55]
[5,6,8]
[1,8,88]
[39,55,100]
[5,6,21,85]
[1,8,39,55,88,100]
[1,5,6,8,8,21,39,55,85,88,100]


Merge - Sort:
0 1 2 3 4 5
mid = 2
Step 1:
4 8 3 1 6 7
Step 2:
4 8 3
mid : 1
1 6 7
mid : 4
: step 9
Step 3:
4 8
3
: Step 7
Step 10
1 6
7
Step 4:
4
8
: Step 5
Step 11:
1
6
12
Step 6:
4 8
13
1 6
3 4 8
: Step 8
1 6 7
15
Step 16:
1 3 4 6 7 8
: sorted array

Q2 Quick - Sort:



MIGHTY PAPER PRODUCT

Quick-sort:

i (or)	Pivot (high)	condition (pivot)	status (True / False)	Array ({Array status})
1	7	(7, 4)	True	{4, 8, 3, 1, 6, 7}
1	7	(7, 8)	false	{4, 8, 3, 1, 6, 7}
2	7	(7, 3)	True	{4, 3, 8, 1, 6, 7}
3	7	(7, 1)	True	{4, 3, 1, 8, 6, 7}
4	7	(7, 6)	True	{4, 3, 1, 6, 8, 7}
4	7	(7, 7)	false	{4, 3, 1, 6, 7, 8}
		Now we call sort of low:0, high=3;		
0	1	(1, 4)	false	{4, 3, 1, 6, 7, 8}
0	1	(1, 3)	false	{4, 3, 1, 6, 7, 8}
1	1	(1, 1)	false	{1, 3, 4, 6, 7, 8}

And after that all the value/sort function return/change the index of value not found.

3) Stack

a.)

Example:-

Q1: Postfix is given "abc*+def*+g*+" we have to find infix and prefix and proved by putting constant 1 value.

Input (Postfix)	Postfix	Infix
abc*+de*f+g*+	g	a
abc*+de*f+g*+	g*+	ab
"	g*+f	a(b*c)
"	g*+fed	a+(b*c)
"	g*+fed*+	a+(b*c)d
"	g*+fed*+a	a+(b*c)de
"	g*+fed*+cba*	a+(b*c)d*e
"	g*+fed*+cba*+	(a+(b*c))(d*e)+f
"	g*+fed*+cba*+	(a+(b*c))((d*e)+f)g

abc*+de*f+g*+ *abc+*feg*d ((a+(b*c))+((dxe)+f))*g)

Calculate:

Put all the value of the varaible 1 such as

∴ a=b=c=d=e=f=g=1;

Infix Notation:

((a+(b*c))+((d*e)+f)*g) ⇒ ((1+1)+((1*1)+1)*1)

((1+1)+((1+1)*1) ⇒ 4

Postfix Notation:

abc*+de*f+g*+ ⇒ (1)(1)(1)*+(1)(1)*(1)+(1)*

= 4

Prefix Notation:

*+abc+*feg*d

= 4

Proved.

(A*B)*C*(D*E)+F

Sol:-

fully partmetrize: (((a*b)*c)*((d+e)+f))

Input	Postfix	Prefix
a*b)*c*(d+f)+f	A	f
*b)*c*(d+E)+f	AB	fE
*b)*c*(d+E)+f	AB*	FED
"	AB*C*	FED+
"	AB*C*D	FED+C
"	AB*C*DE	FED+CB
"	AB*C*DE+	FED+CBA*++
a*b)*c*(d+e)+f	AB*C*DE+f+	++*ABC+DEF

Calculate:~

Infix: (((a*b)*c)*((d+e)+f)) ⇒ (((1*1)*1*1)*((1+1)+1)) ⇒ 3.

Postfix: ab*c*de+f+ ⇒ 11*1*11+1+ ⇒ 3.

Prefix: ***+ABC+DEF ⇒ ++*ABC+DEF ⇒ 3.

4.) Queue:~

Time Complexity:

Insert = O(1)

Space = O(n)

Delete = O(1)

Search = O(n)

Overall Case: (for all the case of insert, delete & search.)

Best Case = O(1)

Avg Case = O(n)

Worst Case = O(n)

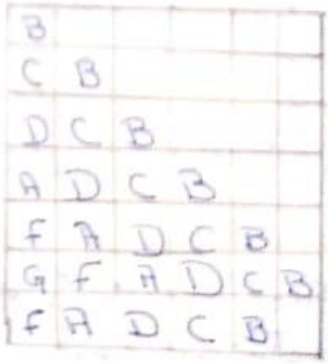
Example 1:

Q A = {1,2,3,4,5,6} and delete two of them.

insert(5);
insert(4);
insert(3);
insert(2);
insert(1);
delete();
delete();
print()

Output: "1,2,3,4"

Q A={B,C,D,A,F,G}



del();
Print().

Output:

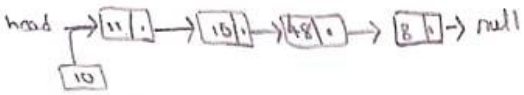
"F,A,D,C,B"

5.) Linklist:~

Single Lisked list:~

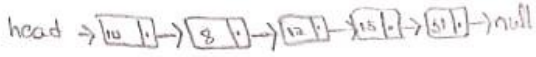
- Insertion:

1. Insert at start:



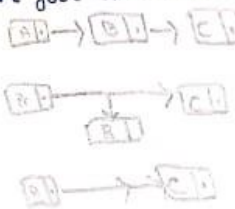
2. Insert at mid.

3. Insert at end



- Deletion:

Same as insert just broke the link between two



- Search:

is simple and linear search, time complexity is O(n) for n size inputs.

Double Linked list:

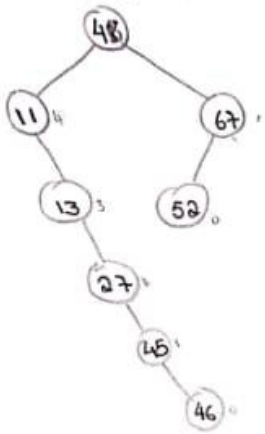
- Insertion: same as just linklist.
- Deletion: same as but just handle the prev point.
- Search: Same as this case just easy to travel the list.

6. Tree :~

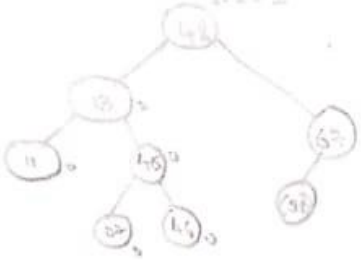
Binary Search tree :~

Example 1:

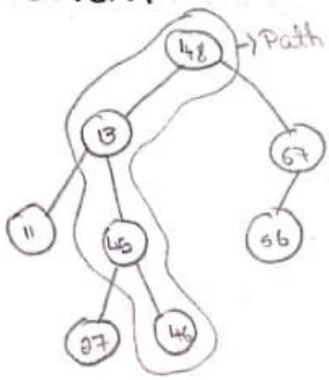
Array = ~~{~~13, 27, 45, 67, 46, 52~~}~~, ~~del~~ Search (46)



L.R rotation :~



Search :

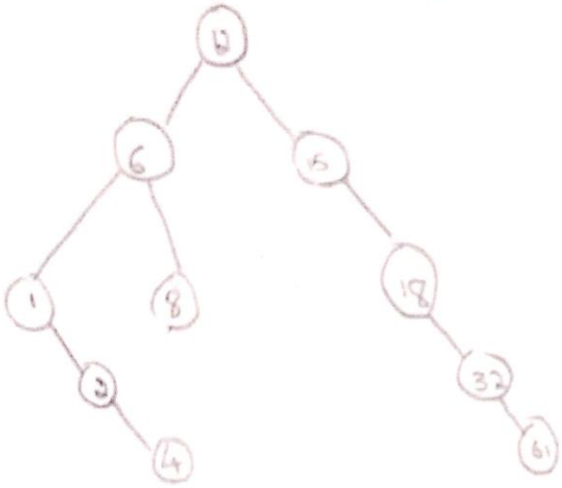


hit = 3hit.

index = 7index.

Example 2:

Array = {12, 6, 8, 15, 18, 32, 61, 1, 2, 4}

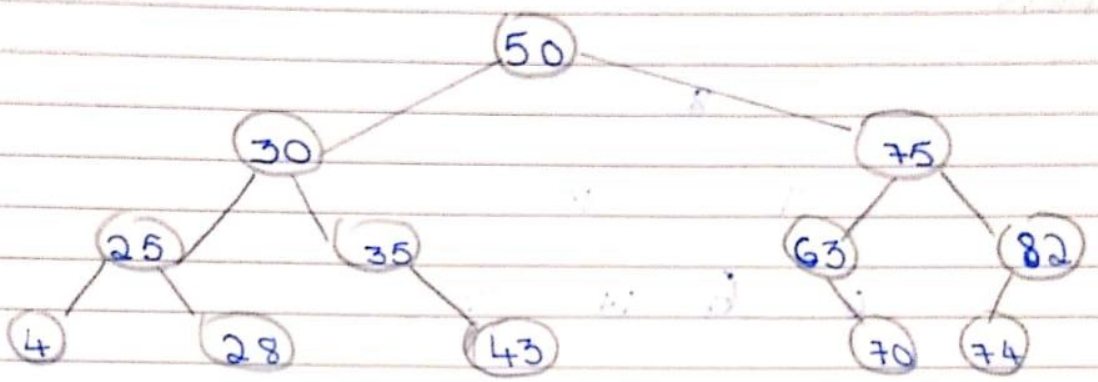


InOrder: 1, 2, 4, 6, 8, 12, 15, 18, 32, 61

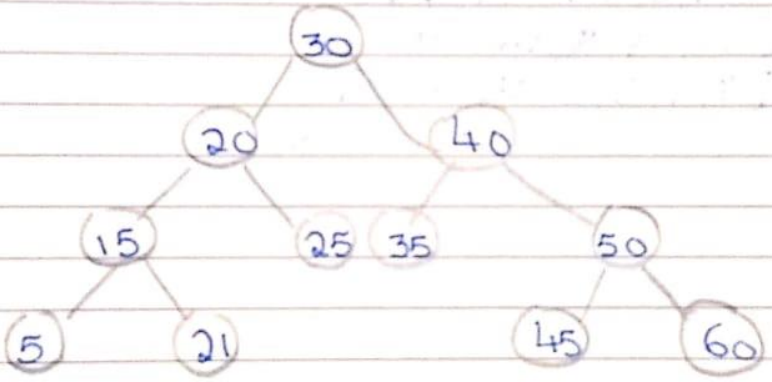
PostOrder: 4, 2, 1, 8, 6, 61, 32, 18, 15, 12

PreOrder: 12, 6, 1, 2, 4, 8, 15, 18, 32, 61

Q2



Q3



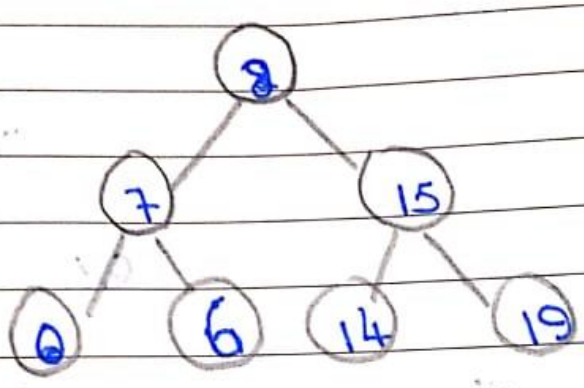
Inorder: 5, 15, 25, 20, 26, 35, 40, 45, 50, 60, 30.

Postorder: 30, 20, 15, 5, 21, 25, 40, 35, 50, 45, 60.

Preorder: 5, 25, 15, 21, 20, 40, 35, 45, 60, 50, 30.

D

Example:-



InOrder: [0, 7, 6, 8, 15, 14, 19]

PostOrder: [0, 6, 7, 14, 19, 15, 8]

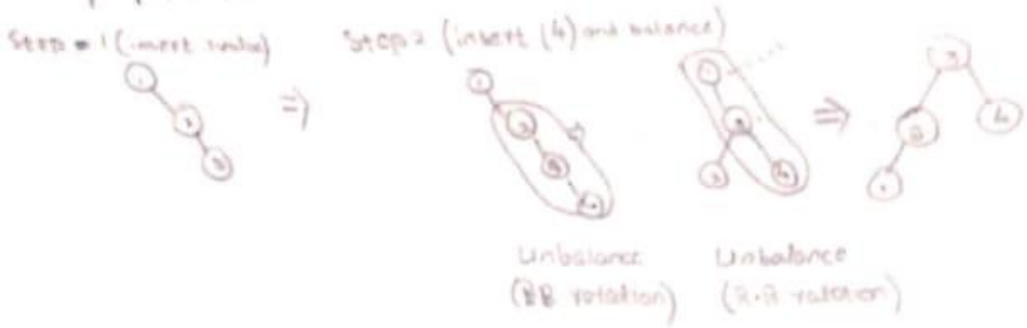
PreOrder: [8, 7, 0, 6, 15, 14, 19]

Ans

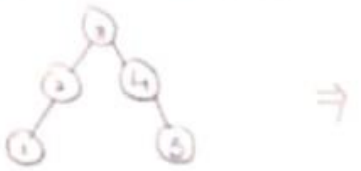
4.) AVL tree:

Example 1:

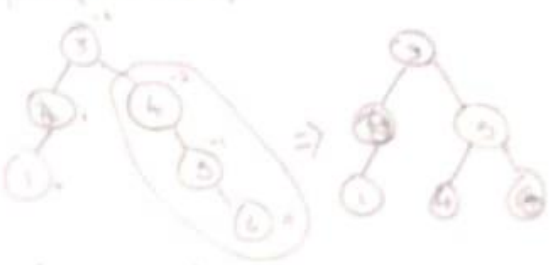
Array = {1,2,3,4,5,6,7}



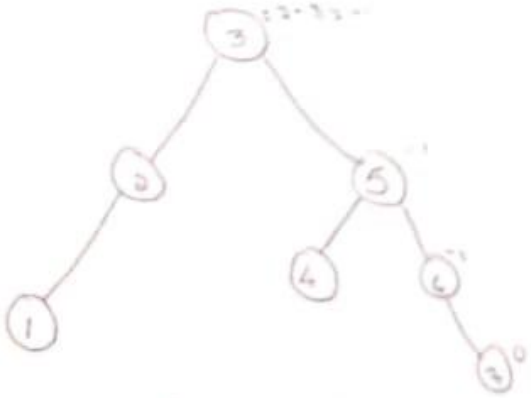
Step = 2 (insert(5))



Step 3 (insert(6))



Step 4: (insert(7))

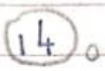


(Balance)

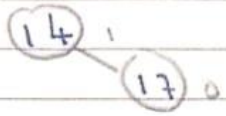
Insertion :

14, 17, 11, 7, 53, 4, 13, 12, 8, 60, 19, 16, 20

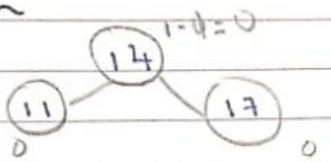
Step #1:-



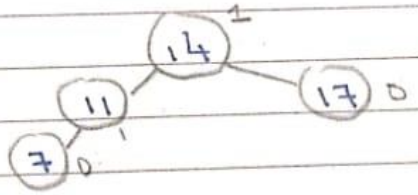
Step #2:-



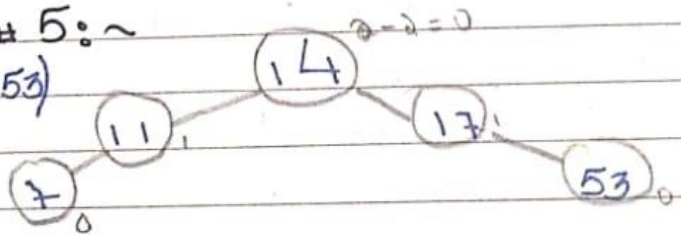
Step #3:-



Step # 4 (insert 7)

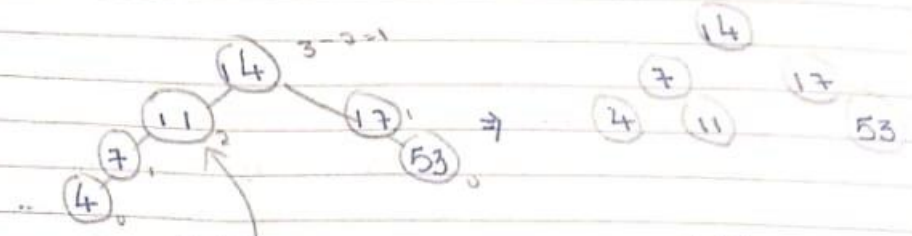


Step # 5:-
(insert 53)

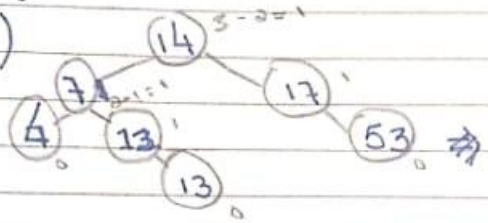


Date:

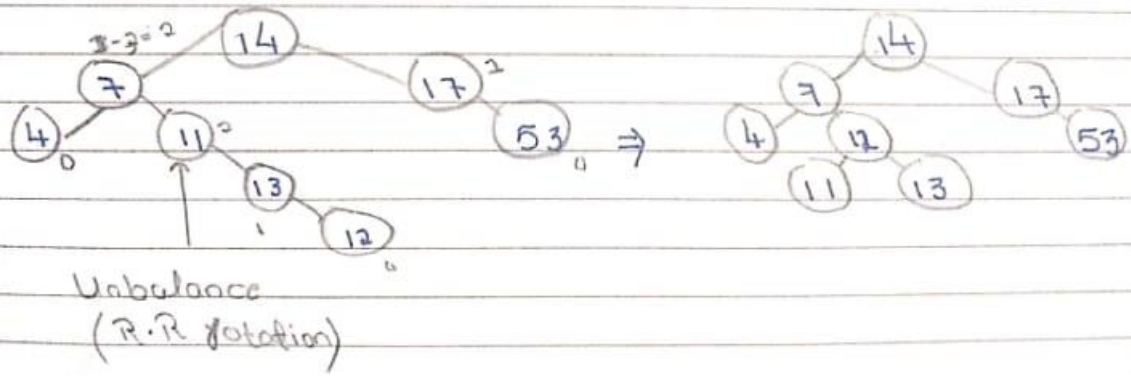
Step#6:~
(insert 4)



Step#7:~
(insert 15)



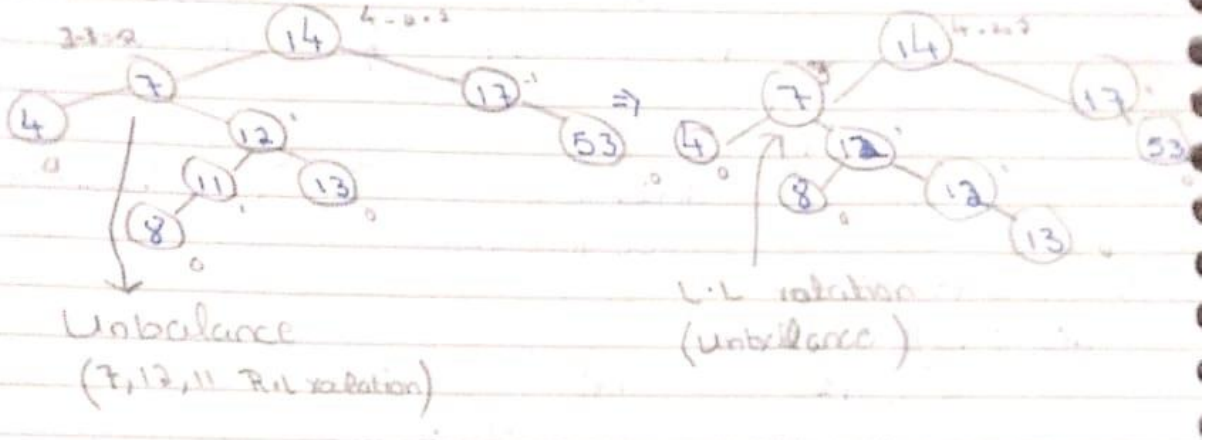
Step#8:~
(insert 12)

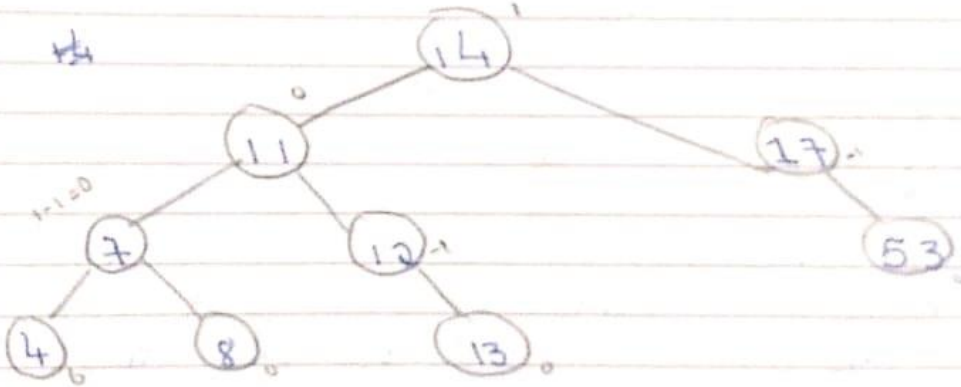


MIGHTY PAPER PRODUCT

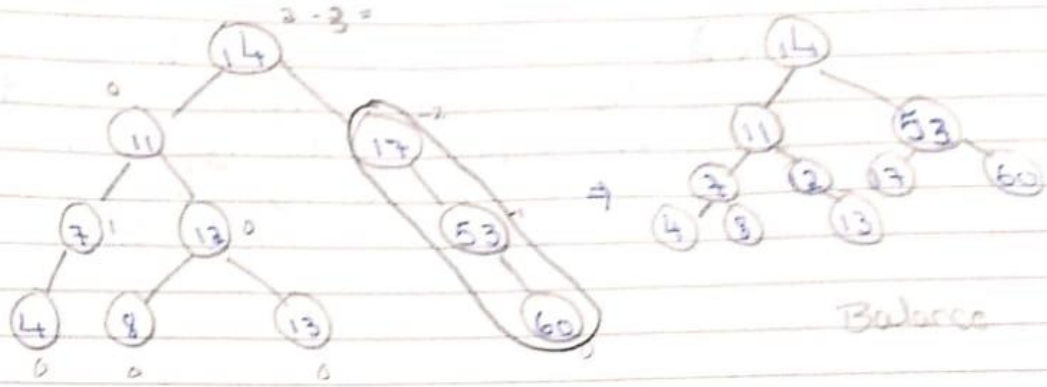
Date:

Step - 9 :-
(insert 8)





Step 10
(insert 60)

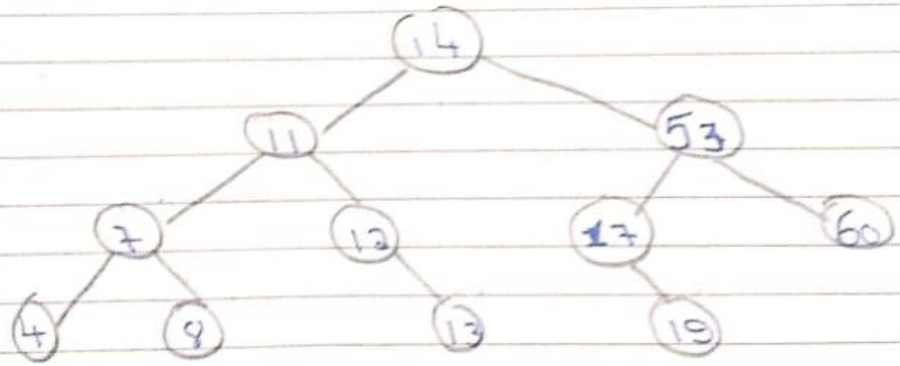


Unbalance

Balance

(R.R rotation)

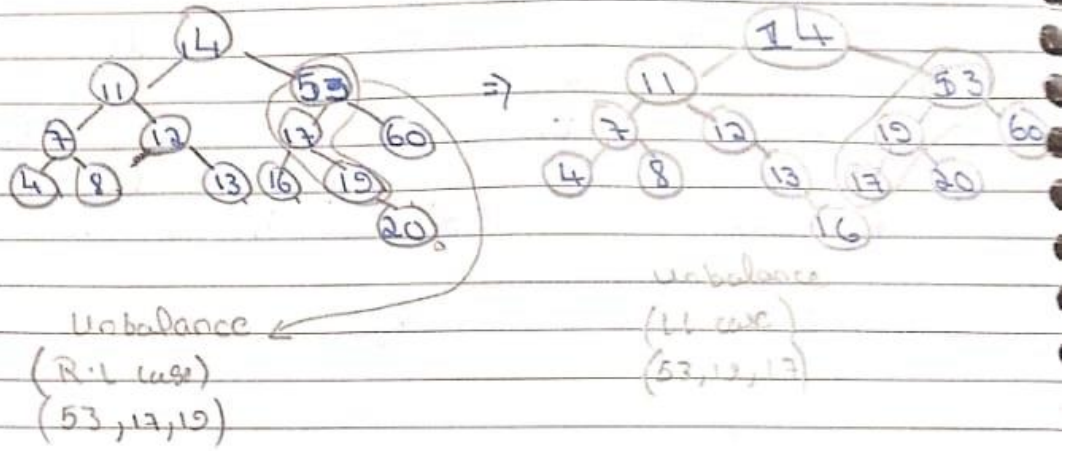
Step 11
(insert 19)



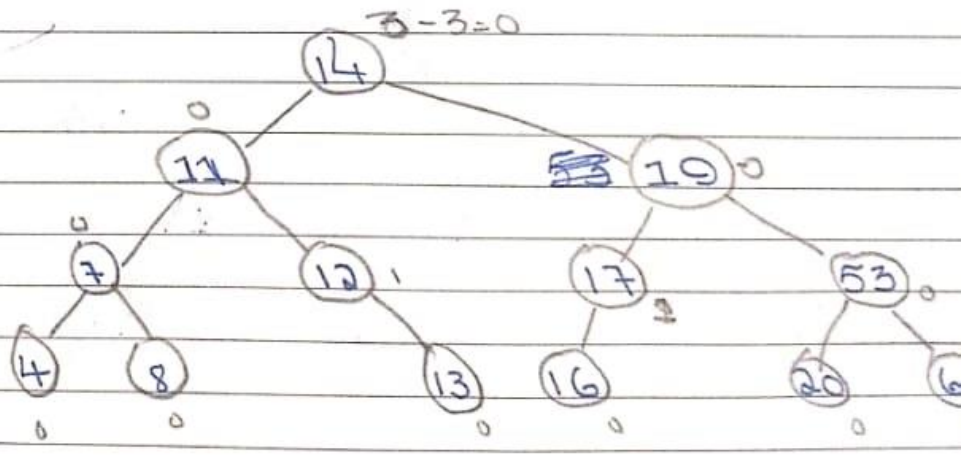
MIGHTY PAPER PRODUCT

Date:

Step #12
(insert 20)



Now final product will be

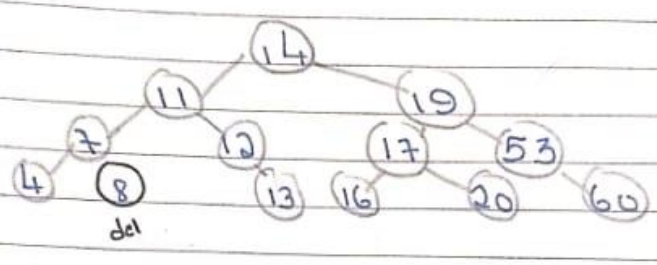


Date:

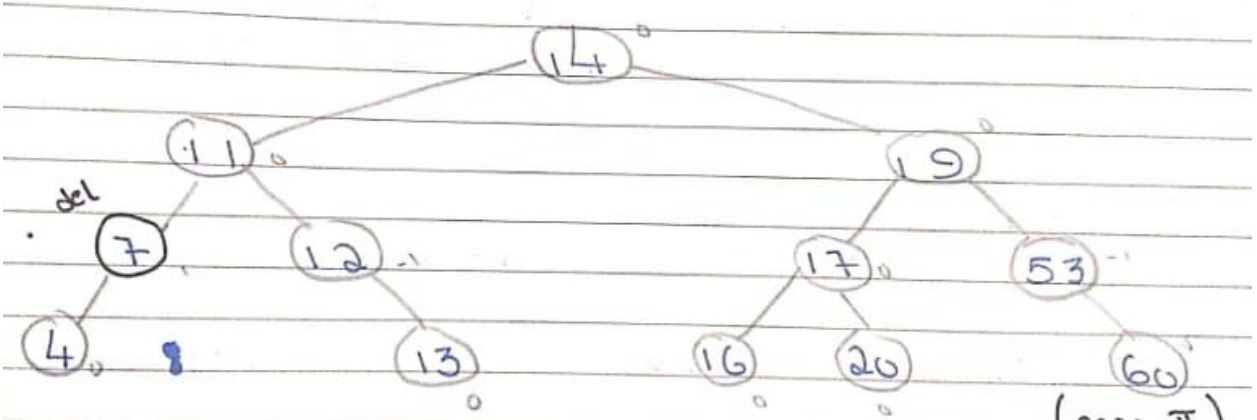
Delete:-

8, 7, 11, 14, 17

Step # 1
(delete 8)

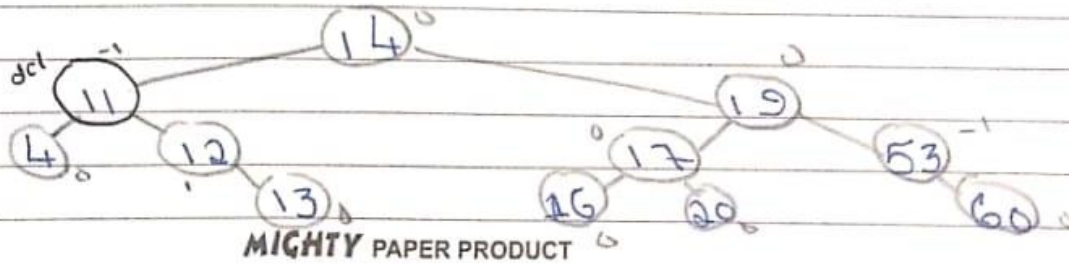


(case I)
leaf root



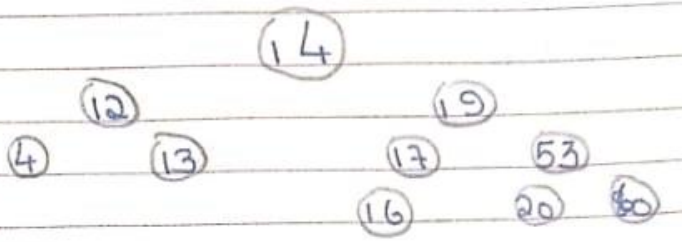
(case II)
Only one chi

(delete 7)



Date:

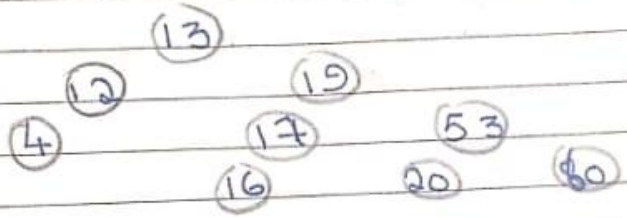
(delete 11)



(del 14)

(Case III)
two child
min of right

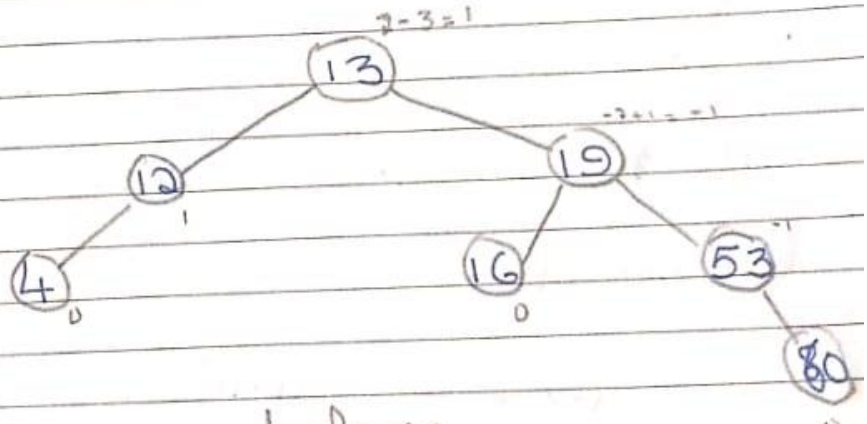
Case III
max from
left become
right.



(del 17)

Case II

Only one
child

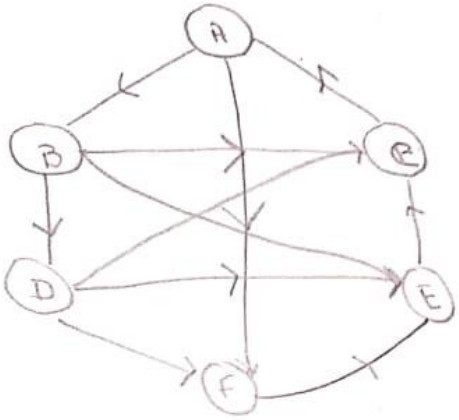


balance

7. Graph:

Representation:-

Example. I:



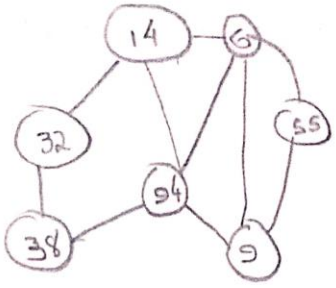
Metrix:

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

Linklist:

A = B, F;
B = C, D, E;
C = A;
D = C, E & F;
E = C;
F = E;

Example 2:



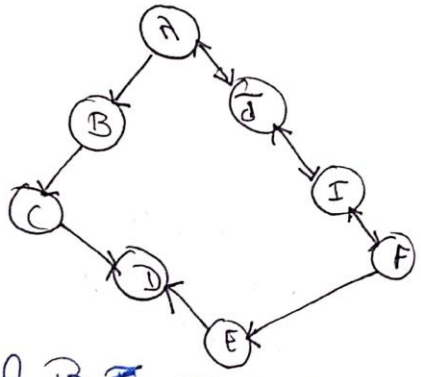
BFS:

14, 32, 94, 38, 9, 55, 6.

DFS:

14, 32, 38, 94, 6, 9, 55.

Example 3:



BFS: A, B, J, C, D, I, f, E.

DFS: A, B, C, D, J, I, f, E.