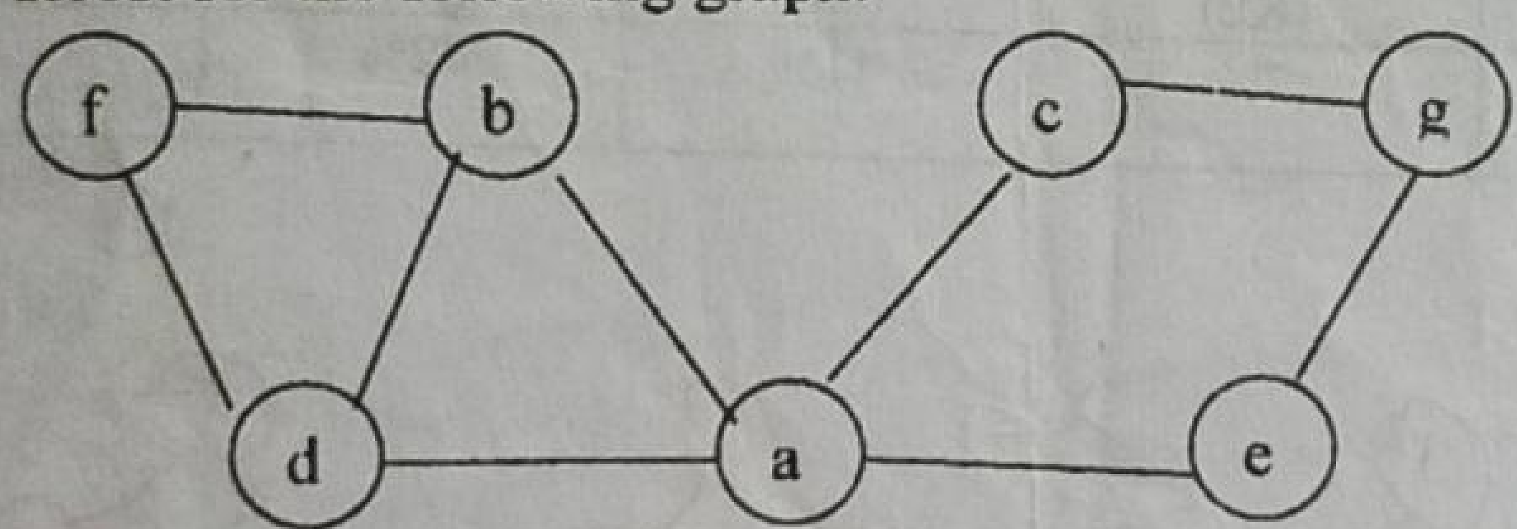
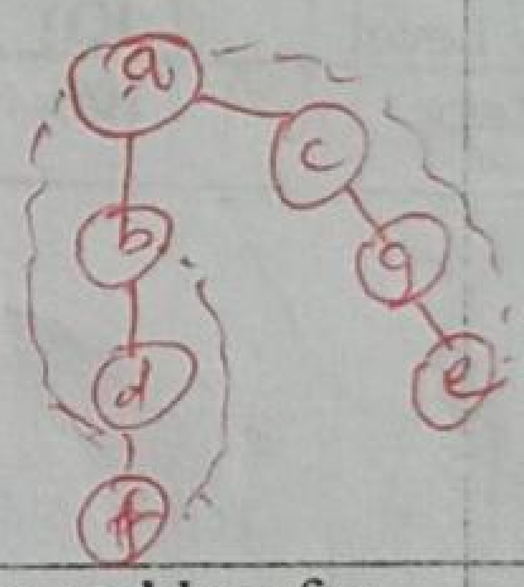
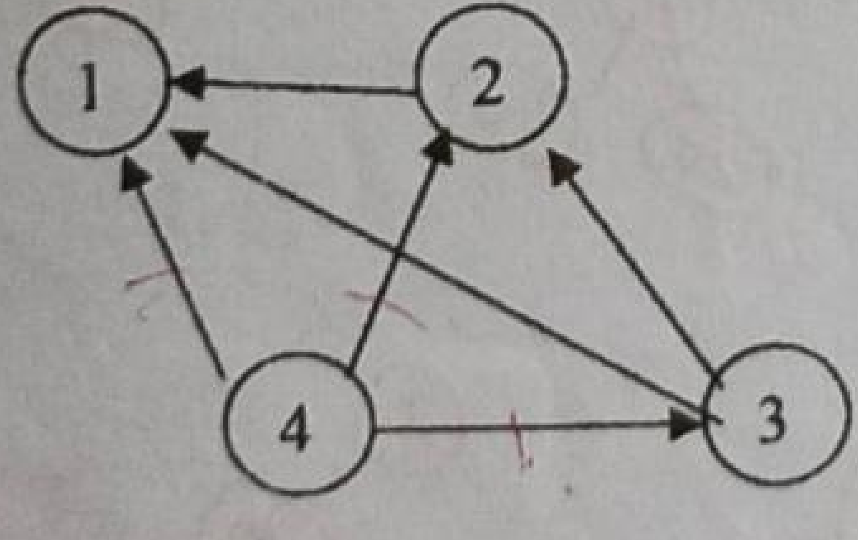
Roll No. | | | | | | | | |

KONGU ENGINEERING COLLEGE, PERUNDURAI 638 060
EVEN SEMESTER 2017-2018
CONTINUOUS ASSESSMENT TEST II - FEBRUARY 2018
(Regulations 2014)

Programme : BE Branch : CSE Semester : IV	Date : 28.02.2018 Time : 9.15 am – 10.45 am
Course Code : 14CST43 Course Name : Design and Analysis of Algorithms	Duration : 1 ½ Hours Max. Marks : 50

PART - A (10 × 2 = 20 Marks)

ANSWER ALL THE QUESTIONS

1.	State Master Theorem.	(CO2)	[K1]
2.	Apply Merge sort to sort the list W,E,L,C,O,M,E.	(CO2)	[K3]
3.	Write an algorithm for finding height of a binary tree.	(CO3)	[K2]
4.	Construct a binary tree with ten nodes labelled 0,1,2,....9 in such a way that the inorder and post order traversals of the tree yield the following lists: In order : 9,3,1,0,4,2,7,6,8,5 Post order: 9,1,4,0,3,6,7,5,8,2	(CO2)	[K3]
5.	List the three major variations of Decrease-and-Conquer approach.	(CO3)	[K1]
6.	Find DFS forest for the following graph:  	(CO2)	[K3]
7.	Apply the Source removal algorithm to solve the topological sorting problem for the following digraph:  4 4, 3, 2, 1	(CO3)	[K3]
8.	Write an algorithm to solve element uniqueness problem using Presorting.	(CO3)	[K2]
9.	Mention the advantages of balanced search tree.	(CO3)	[K1]
10.	Computer 2101 * 1130 by applying the Divide-and-Conquer algorithm.	(CO2)	[K3]

2101 × 1130

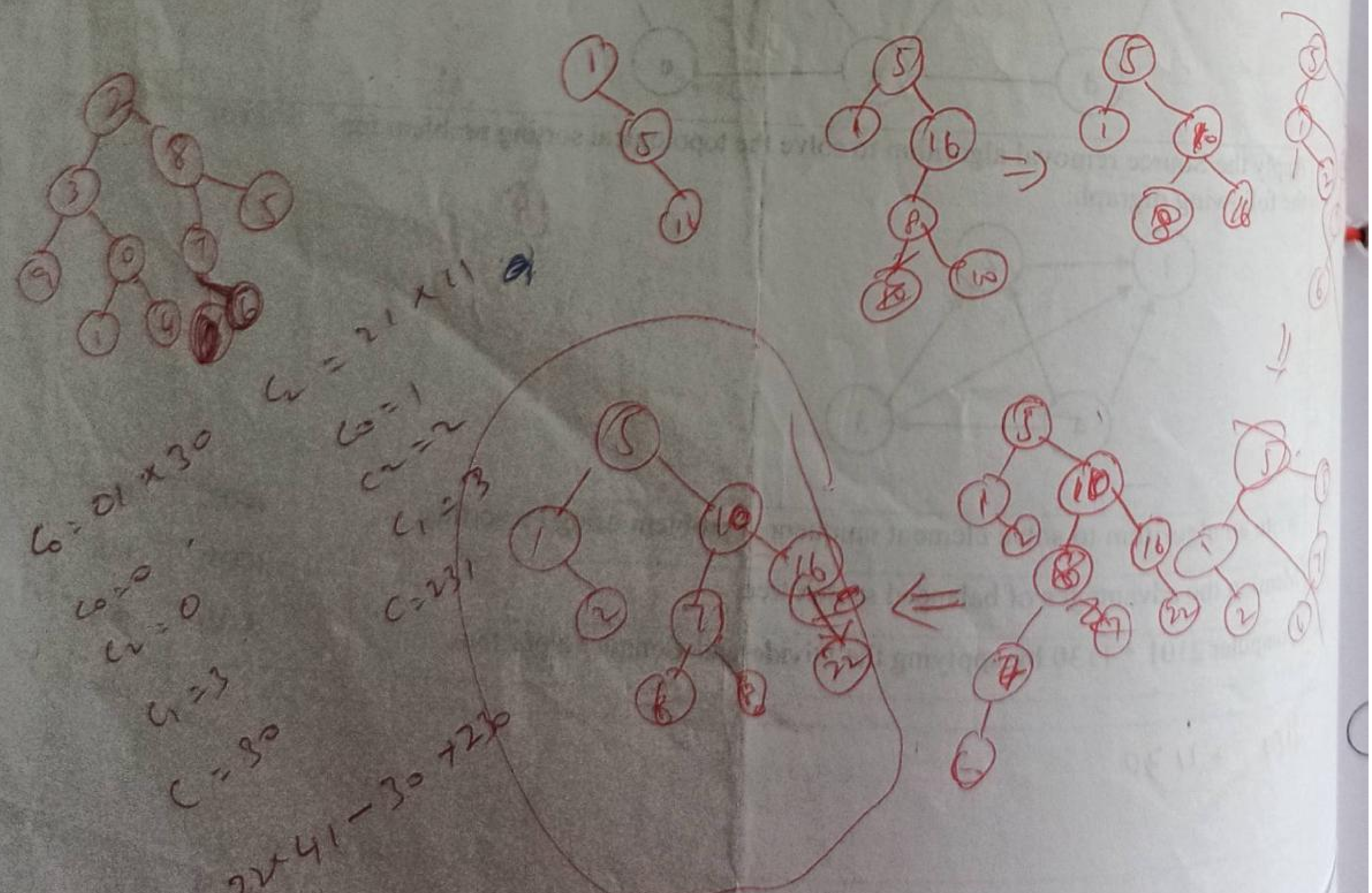
check adjacent element

4

Scanned by TapScanner

Part – B (3× 10 = 30 Marks)					
ANSWER ANY THREE QUESTIONS					
11.		Write an algorithm for Quick sort. Also analyze its average case time complexity.	(10)	(CO2)	
12.	i)	Write an algorithm for insertion sort to arrange the list of n numbers in ascending order and analyze its time complexity.	(5)	(CO3)	
	ii)	Apply Strassen's Matrix multiplication method to multiply the two matrices $\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix}\begin{pmatrix}4 & 3\\2 & 1\end{pmatrix}$	(5)	(CO2)	
13.		Consider the problem of finding the smallest and largest elements in an array of n numbers. a)Design a presorting–based algorithm and determine its efficiency class. b)Compare the efficiency of the following three algorithms for the above problem: i) Brute force algorithm n ii) Presorting–based algorithm – nlog n + n iii) Divide-and-conquer algorithm n log n	(10)	(CO3)	
14.		Illustrate various types of rotations in AVL tree with an example. Construct AVL tree for the list 1, 5, 16, 8, 10, 2, 22, 7, 6, 1 by successive insertions.	(10)	(CO3)	

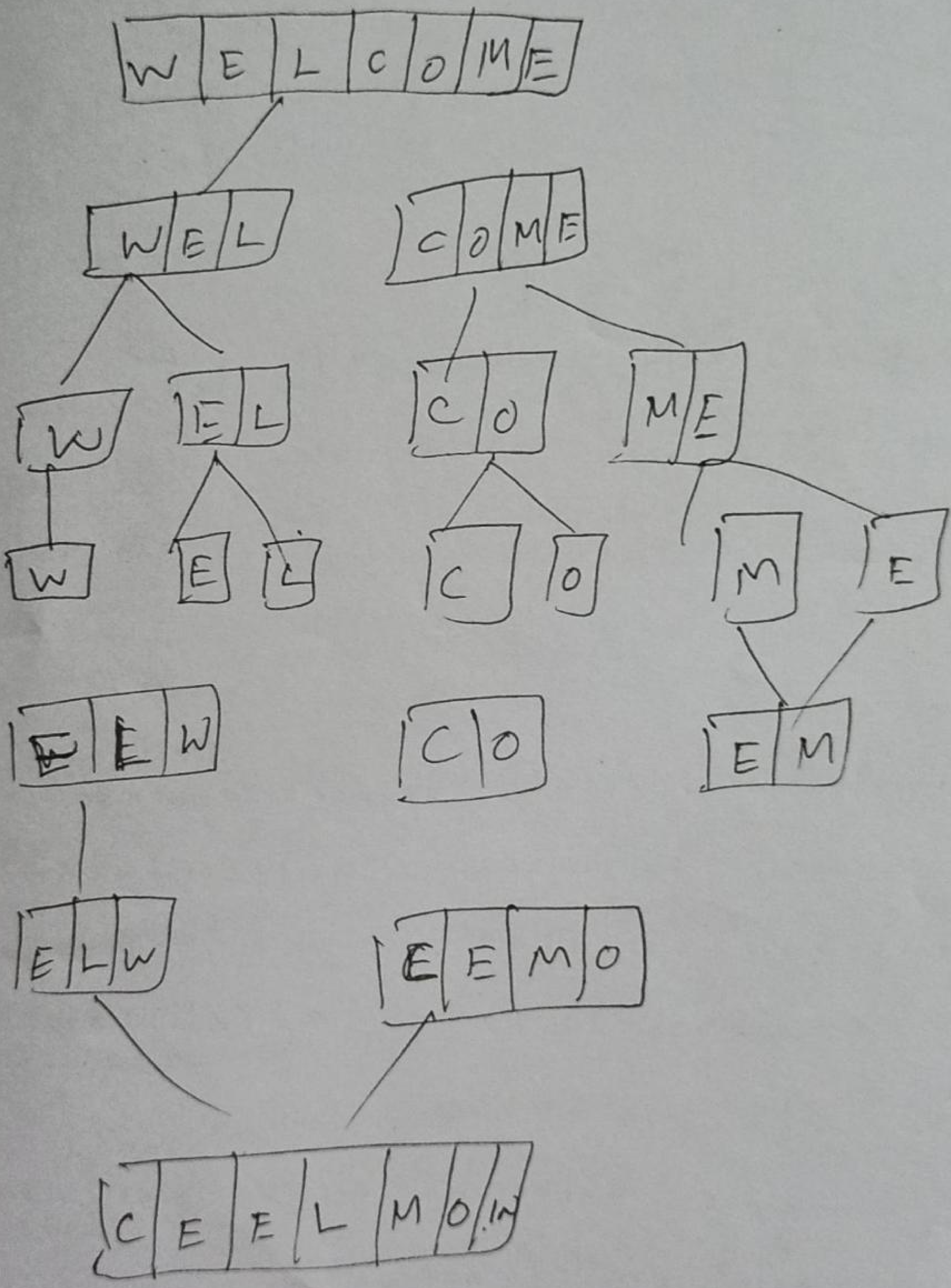
Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	10%	7%	41%	42%	-	-



$C_0 = 01 \times 30$
$C_2 = 2^1 \times 21$
$C_0 = 1$
$C_2 = 2$
$C_1 = 3$
$C = 231$
$C_0 = 0$
$C_2 = 0$
$C_1 = 3$
$C = 30$
$C_1 = 22 \times 41 - 30 + 2$
641
$C = 2379130$

Scanned by TapScanner

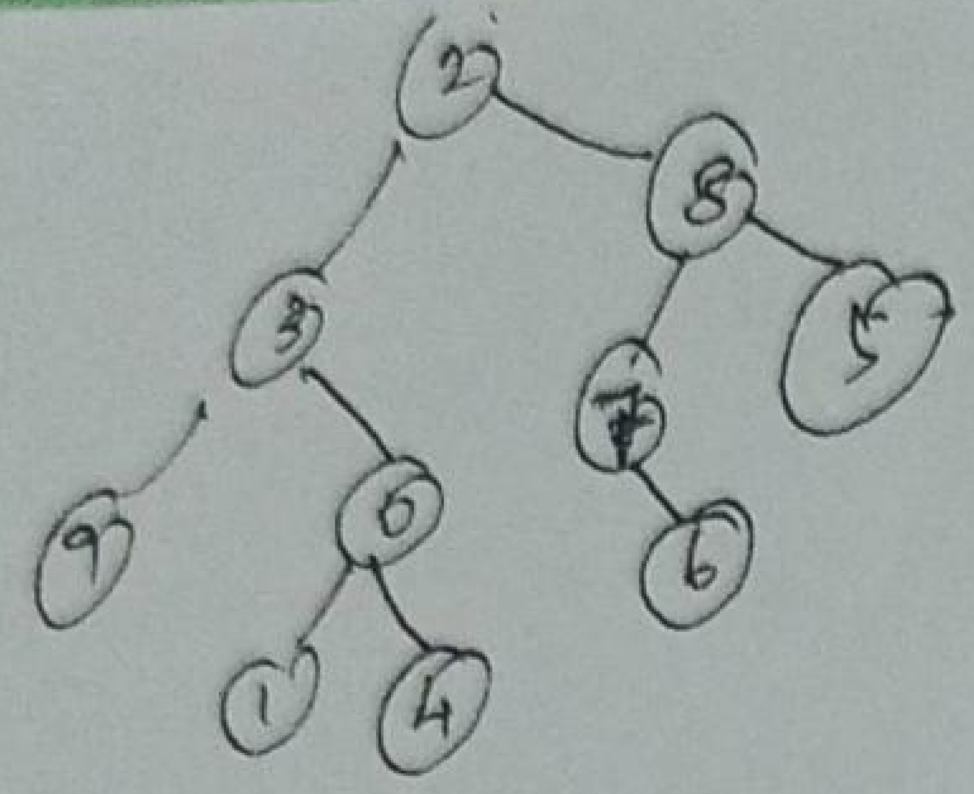
1. $$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{if } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

2. 

3. if T = ∅ return −1
else return max{height(T_L), height(T_R)} + 1.

4.

(4)

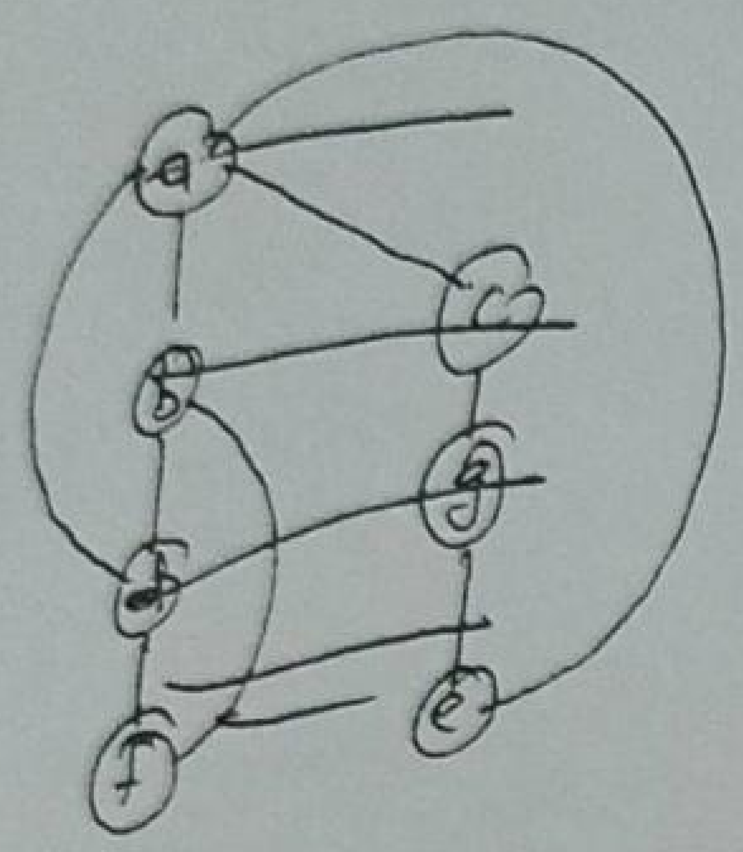


(5) (i) decrease by constant

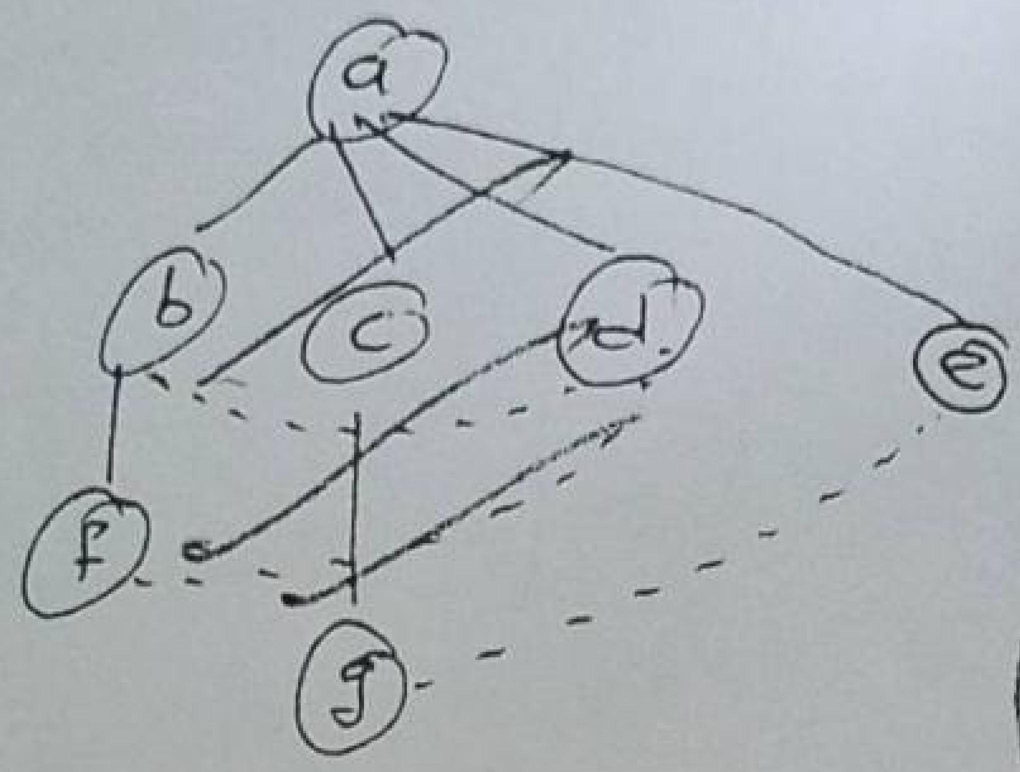
(ii) decrease by constant fac

(iii) ~~d~~ variable size decrease

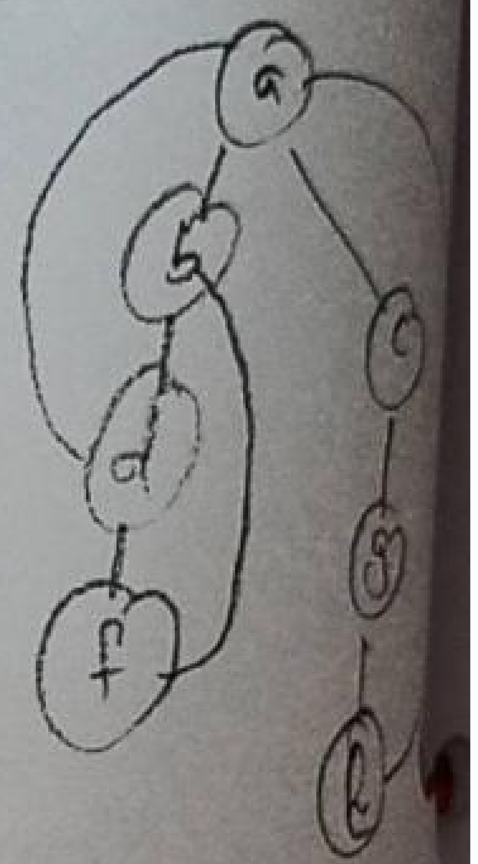
(b)



(6)



(7) A→5



(9)

(8) for $i \leftarrow 0$ to $n-2$ do

if $A[i] = A[i+1]$ return false

return true

(10) 2101×1130

$c_2 = 21 \times 11$

$c_0 = 01 \times 30$

$c_1 = (21+01) * (11+30) - (c_2 + c_0) = 22 \times 41 - 21 \times 11 - 01 \times 30$

$\underline{21 \times 11}$ $c_2 = 2 \times 1 = 2$ $c_0 = 1 \times 1 = 1$ $c_1 = (2+1) * (1+1) - (2+1) = 3$

So $21 \times 11 = 2 \times 10^2 + 3 \cdot 10^1 + 1 = 231$

$\underline{01 \times 30}$ $c_2 = 0 \times 3 = 0$ $c_0 = 1 \times 0 = 0$ $c_1 = (0+1)(3+0) - 0 = 3$

So $01 \times 30 = 0 \cdot 10^2 + 3 \cdot 10^1 + 0 = 30$

$22 \times 41 =$ $c_2 = 2 \times 4 = 8$ $c_0 = 2 \times 1 = 2$ $c_1 = (2+2) * (4+1) - (8+2) = 10$

So $22 \times 41 = 8 \cdot 10^2 + 10 \cdot 10^1 + 2 = 902$

$2101 \times 1130 = 231 \cdot 10^4 + (902 - 231 - 30) \cdot 10^2 + 30 = 2,374,130.$

Alg Quicksort (A [l..r])

if l < r .

S ← Partition (A[l..r])

Quicksort(A [l..s -1])

Quicksort (A [s+1...r])

Alg partition(A[l..r])

P ← A[l]

i ← l; j ← r+1

repeat

repeat i ← i+1 until A [i] ≥ P

repeat j ← j-1 until A[j] ≤ P

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

until i ≥ j

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

swap (A[l], A[j])

return j

Avg Case Analysis:

$$\begin{cases} C_{avg} = \frac{1}{n} \sum_{s=0}^{n-1} [(n+1) + C_{avg}(s) + C_{avg}(n-1-s) & n \geq 1 \\ C_{avg}(0) = 0 \\ C_{avg}(1) = 0 \end{cases}$$

~~Sb~~

~~Cavg~~.

C(n) = ~~R~~ 38 $\log_2 n$.

. 2 (i) Insertion Sort:

Alg insertionSort(A[0..n-1])

for i ← 1 to n-1 do

v ← A[i]

j ← i-1

while j ≥ 0 & A[j] > v do

A[j+1] ← A[j]

j ← j-1

done

A[j+1] ← v

Time Complexity:

$$C_{Worst}(n) = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} 1$$

$$= \frac{n(n-1)}{2} = \underline{\underline{O(n^2)}}$$

$$C_{Best} = \sum_{i=1}^{n-1} 1 = \underline{\underline{O(n)}}$$

Avg: more or less WorstCase

$$C_{avg}(n) = \underline{\underline{O(n^2)}}$$

(4) (9) (8) (15) (ii) 1.

12 (ii) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ e & d \end{pmatrix}$ $\begin{pmatrix} 4 & 3 \\ 2 & 1 \\ g & h \end{pmatrix}$

$P_1 = a(f-h)$ $\qquad$ $P_1 = 1(3-1) = 2$

$P_2 = \cancel{(a+b)\cdot h}\ (a+b)\cdot h$ $\qquad$ $P_2 = (1+2)\times 1 = 3$

$P_3 = (c+d)e$ $\qquad$ $P_3 = (3+4)4 = 28$

$P_4 = d(g-e)$ $\qquad$ $P_4 = 4(2-4) = -8$

$P_5 = (a+d)(e+h)$ $\qquad$ $P_5 = (1+4)(4+1) = 25$

$P_6 = (b-d)(g+h)$ $\qquad$ $P_6 = (2-4)(2+1) = -6$

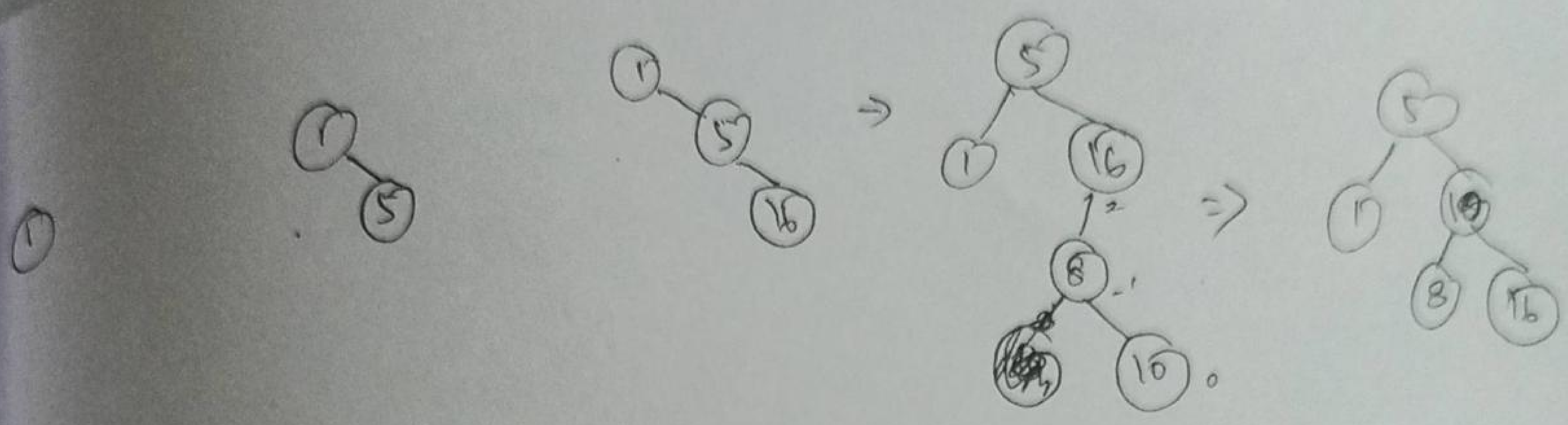
$P_7 = (a-c)(e+f)$ $\qquad$ $P_7 = (1-3)(4+3) = -14$

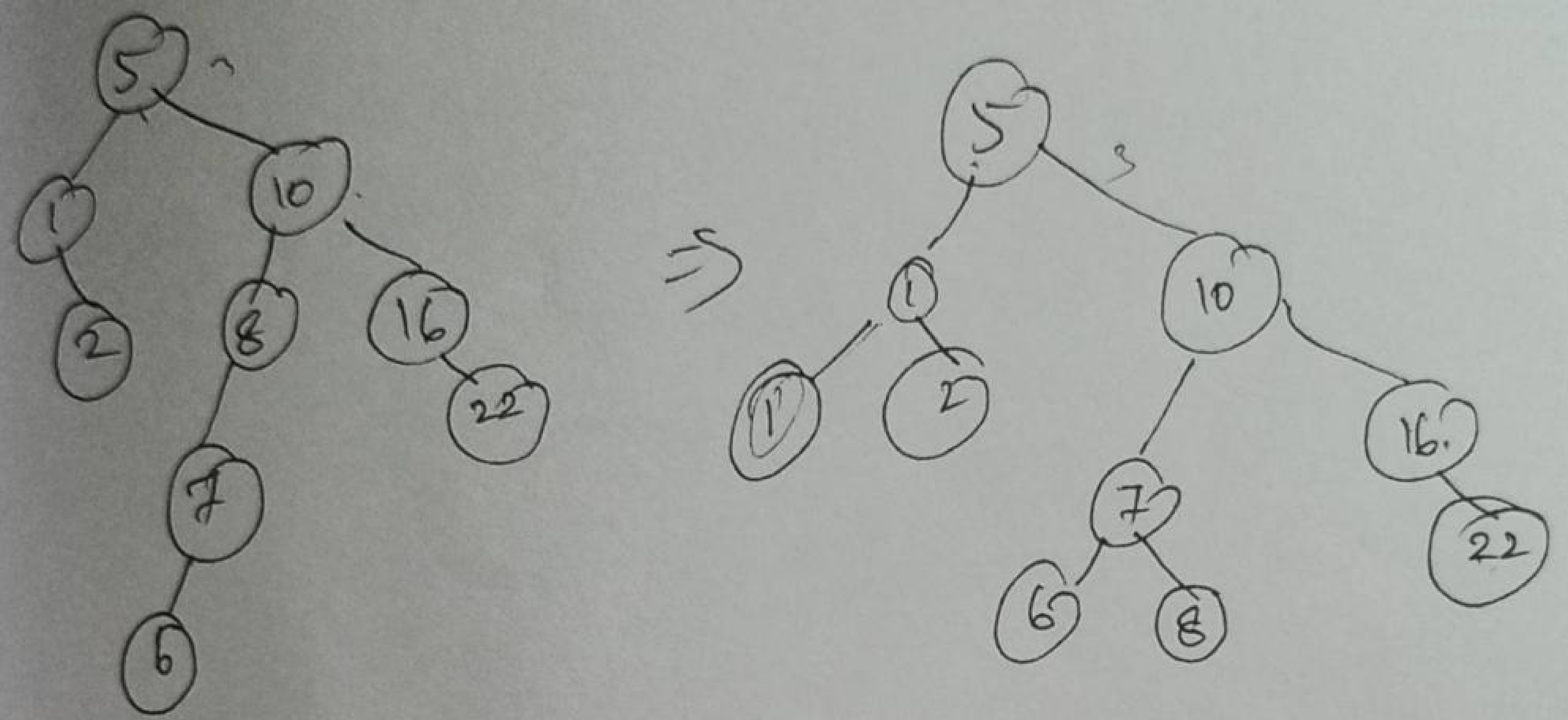
$$= \begin{bmatrix} P_5+P_4-P_2+P_6 & P_1+P_2 \\ P_3+P_4 & P_1+P_5-P_3-P_7 \end{bmatrix}$$

$$= \begin{bmatrix} 25-8-3-6 & 5 \\ 20 & 2+25-28+14 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

13

Scanned by TapScanner





5

Dr. R. MANJULA Devi	28/2/18
Name and Signature of Hall Supdt. with Date	

KONGU ENGINEERING COLLEGE

PERUNDURAI ERODE - 638 060.

(Autonomous)

Estd : 1984

Name of the Student	B.GOKULNATH	Register No.	1	6	C	S	R	0	6	4	
Programme	BE	Branch & Semester	CSE IV								
Course Code and Name	14CST43 Design and Analysis of Algorithms	Date	28/2/18	No. of Pages Used	10						

MARKS TO BE FILLED IN BY THE EXAMINER

PART - A		PART - B			Grand Total Max. Marks : 50
Question No.	Max Marks : 2	Question No.		Max Marks : 10	
1	2	11	i)	9	46 4/50
2	2		ii)		Verified B. Gokulnath
3	0	12	i)	10	
4	2		ii)		
5	2	13	i)		
6	2		ii)		
7	2	14	i)	10	
8	2		ii)		
9	2				
10	1				
TOTAL	17	TOTAL		29	
Total Marks in Words : four six					

INSTRUCTION TO THE CANDIDATE

1. Check the Question Paper, Programme, Course Code, Branch Name etc., before answering the questions.
2. Use both sides of the paper for answering questions.
3. POSSESSION OF ANY INCRIMINATING MATERIAL AND MALPRACTICE OF ANY NATURE IS PUNISHABLE AS PER RULES.

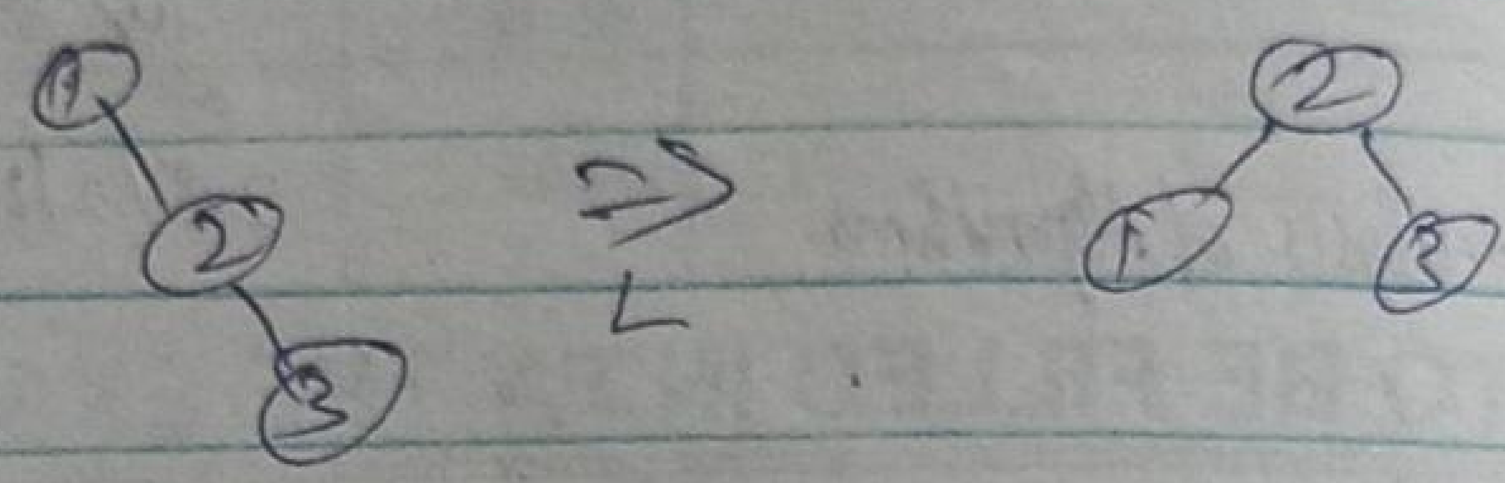
Name of the Examiner

Signature of the Examiner with Date 28/2/18

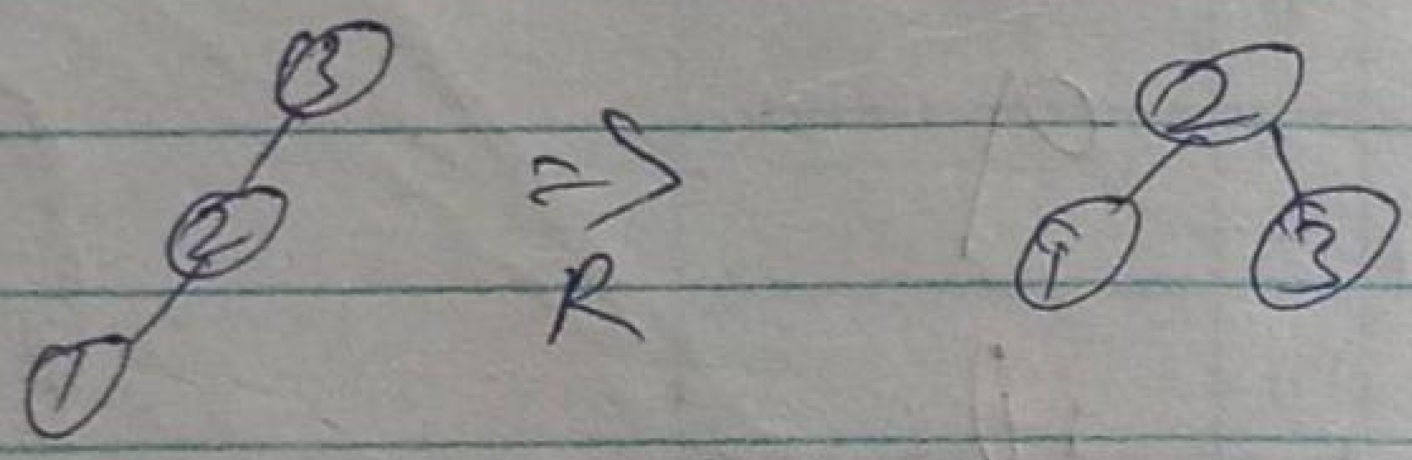
PART-B

14) Various types of rotation in AVL tree:

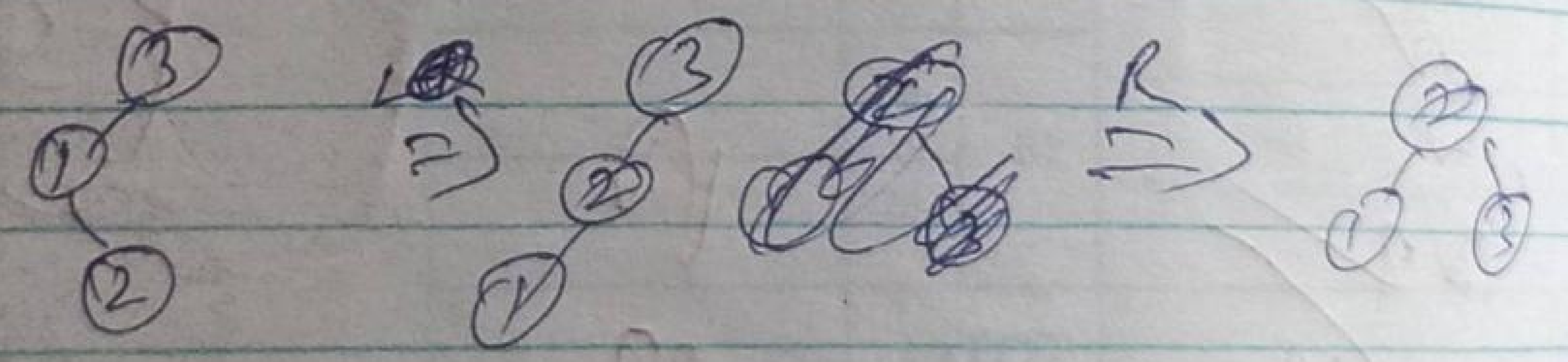
(i) L rotation on single left rotation:



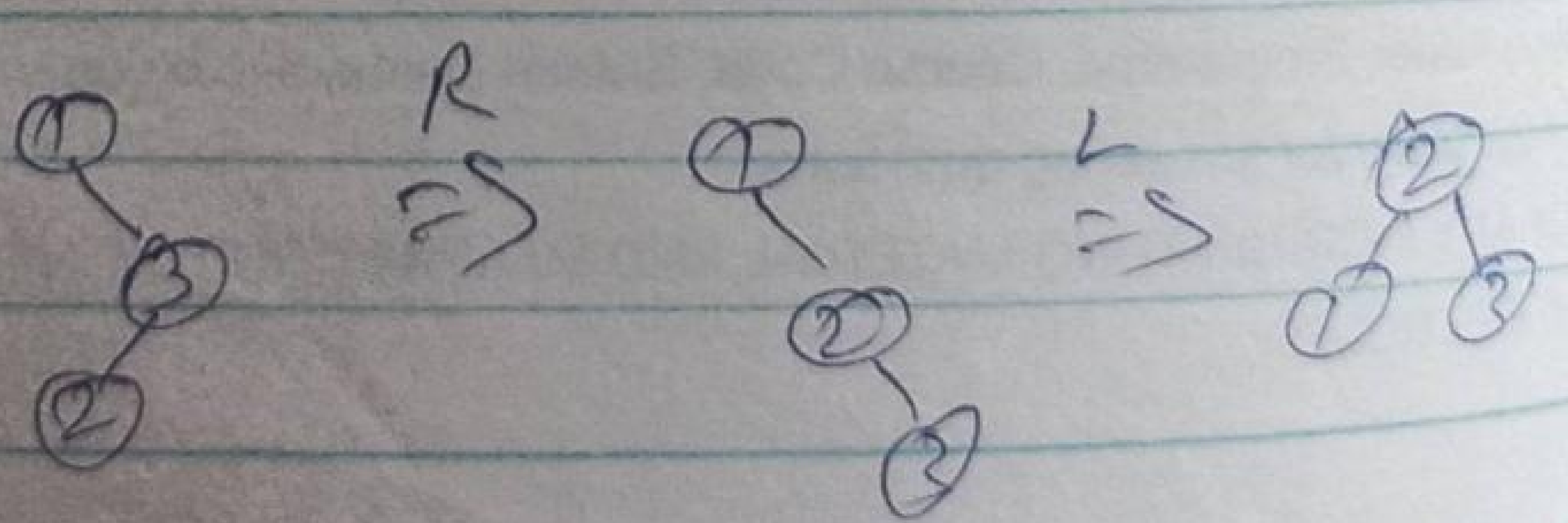
(ii) R rotation (or) single Right rotation:



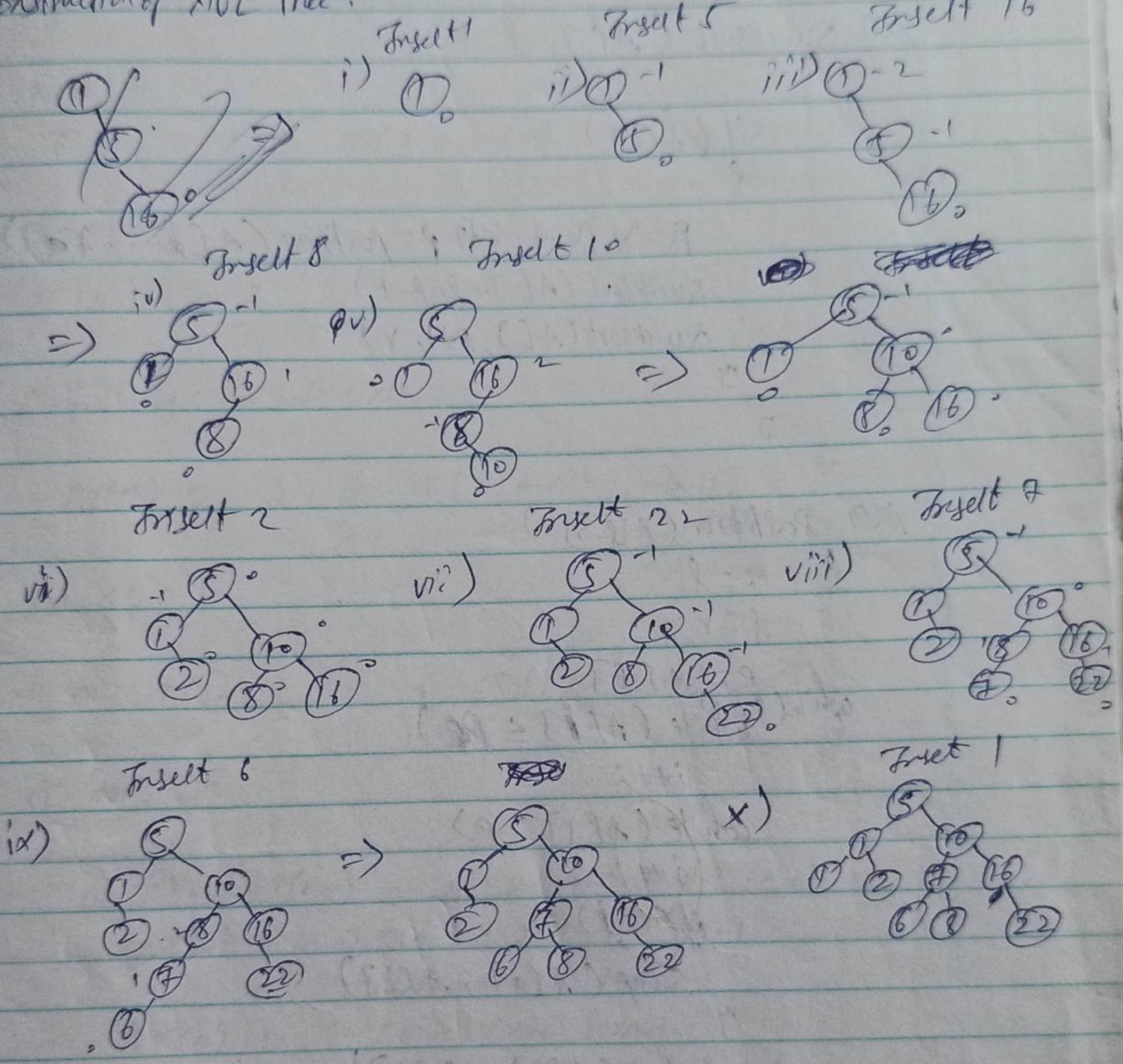
(iii) L-R rotation in Double rotation:



(iv) R-L rotation in Double rotation:



Construction of AVL tree:



10

11)

```
Alg Quicksort (A[ ], l, r)
{
    if (l < r)
    {
        p = partition (A [l, .... r ))
        quicksort (A[ ], l, p-1)
        Quicksort (A[ ], p+1, r)
    }
}

Alg partition (A [l...r] )
{   i = l;
    j = r;
    p = A [l] .
while( i < j )
{   while (A[i] ≤ p )
        i++;
    while (A[j] > p)
        j++;
    if (i < j)
        swap (A[i] , A[j])
}
    swap (A[j] , A[l])

        return j;
}
```

Scanned by TapScanner

Average case time complexity: → Recc eqn ?

$$T(n) = \frac{1}{n}\sum_{j=0}^{n-1} T(j) + \frac{1}{n}\sum_{j=0}^{n-1} T(j) + n$$

$T(1) = 0$

$$T(n) = \frac{2}{n}\sum_{j=0}^{n} T(j) + n \rightarrow ①$$

$T(n-i)$?

Multiply by n on both sides,

$$nT(n) = 2\sum_{j=0}^{n} T(j) + n^2 \rightarrow ②$$

Put $n = n-1$

$$(n-1)\,T(n-1) = 2\sum_{j=0}^{n-1} T(j) + (n-1)^2 \rightarrow ③$$

② - ③ ⇒

$$nT(n) - (n-1)T(n-1) = n^2 - (n-1)^2 + 2T(n-1)$$

$$nT(n) = (n-1)\,T(n-1) + 2T(n-1) + 2n-1$$

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

$$nT(n) = T(n-1)(n+1) + 2n-1$$

neglect ; 1

Divide by $n(n+1)$

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

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

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

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

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

For general, $n=4$

$$\frac{T(n)}{n+1} = \frac{T(1)}{2} + 2\left[\frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right]$$

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

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

$$\frac{T(n)}{n+1} \approx 2\log n$$

$$T(n) = 2\log n\,(n+1)$$

$$T(n) \approx n\log n$$

$$T(n) \in O(n\log n)$$

---

12)

ii)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 4 & 3 \\ 2 & 1 \end{pmatrix}$$

Scanned by TapScanner

COMPUTER SCIENCE & ENG...
...ING COLLEGE,

$S_1 = 3 - 1 = 2$ $\quad P_1 = 1 \cdot 2 = 2$

$S_2 = 1 + 2 = 3$ $\quad P_2 = 3 \cdot 1 = 3$

$S_3 = 3 + 4 = 7$ $\quad P_3 = 7 \cdot 4 = 28$

$S_4 = 2 - 4 = -2$ $\quad P_4 = 4 \cdot -2 = -8$

$S_5 = 1 + 4 = 5$ $\quad P_5 = 5 \cdot 5 = 25$

$S_6 = 4 + 1 = 5$ $\quad P_6 = -2 \cdot 3 = -6$

$S_7 = 2 - 4 = -2$ $\quad P_7 = -2 \cdot 7 = -14$

$S_8 = 2 + 1 = 3$

$S_9 = 1 - 3 = -2$

$S_{10} = 4 + 3 = 7$

$$C = \begin{bmatrix} -3-8+25-6 & 2+3 \\ 28-8 & 2-28+25+14 \end{bmatrix}$$

$$C = \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

i) Insertion sort:

```
Alg insertion sort (A[ ], n)
{
    for i=1 to n-1
    {   v = A[i+1]
        j = i
        while( j≥0 && A[j] > v)
        {     A[j+1] = A[j]
              j--;
        }
        A[j] = v;
    }
}
```

Time complexity :

For worst / average case

$$T(n) = \sum_{i=1}^{n-1} \sum_{j=0}^{i} j$$

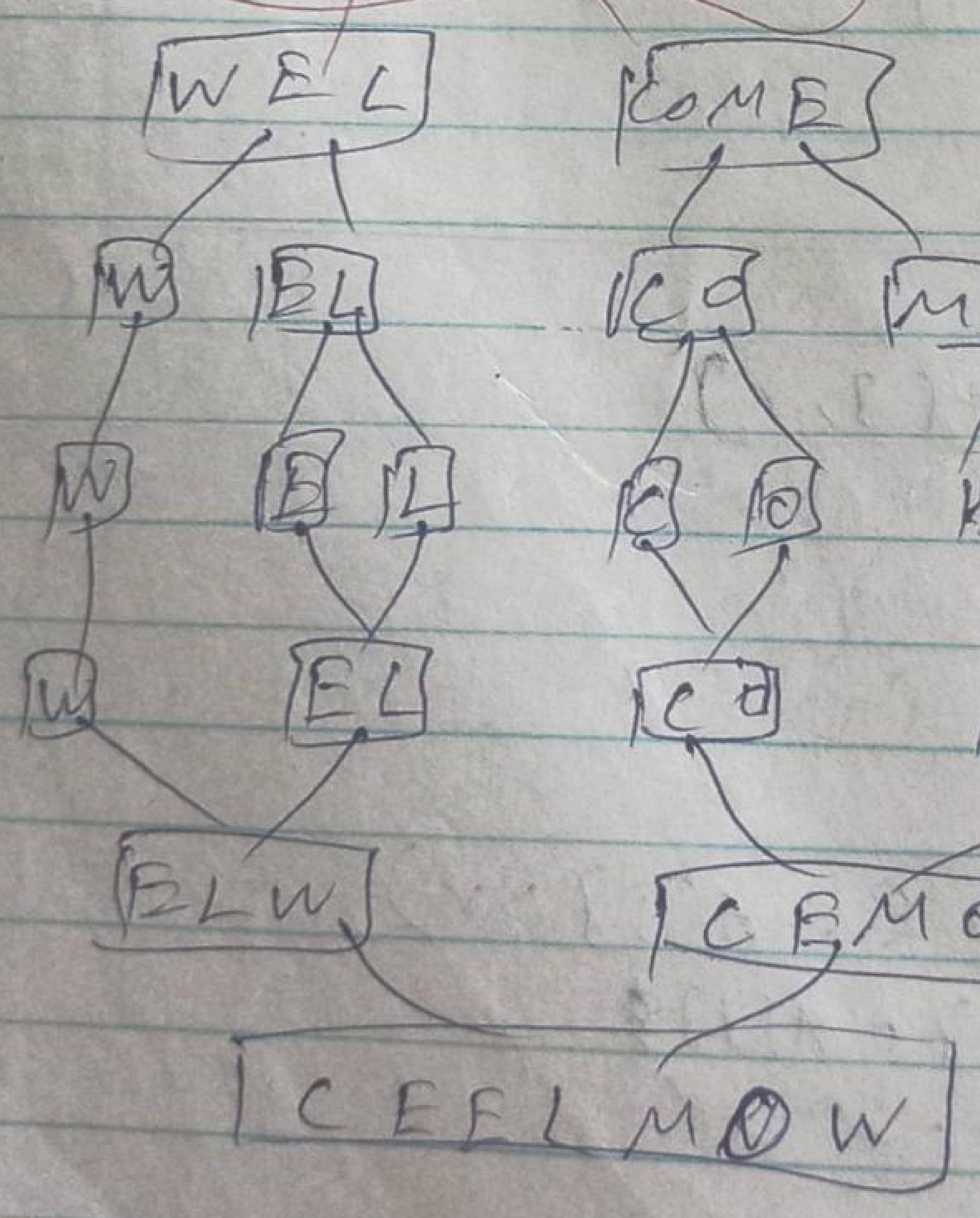
$$= \frac{n(n+1)}{2}$$

$$\simeq O(n^2)$$

For best case, all the elements in ascending order itself, $O(n)$

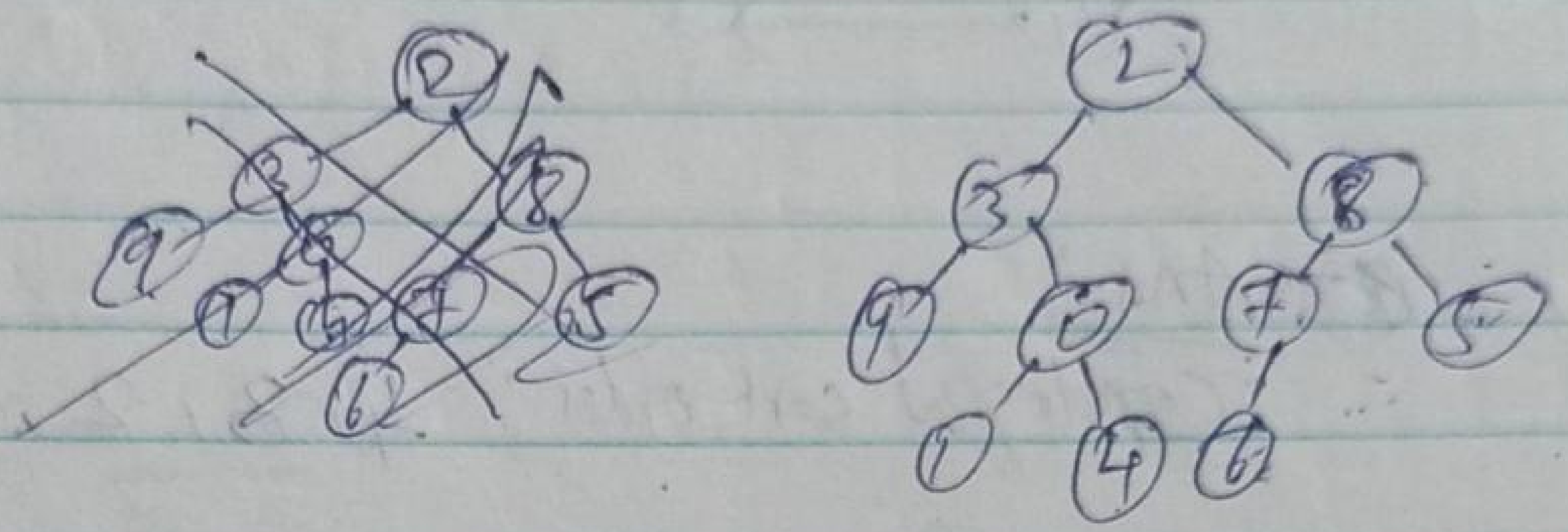
PART-A

2) W E L C O M E



4)

Inorder: 9, 3, 1, 0, 4, 2, 7, 6, 8, 5.
postorder: 9, 1, 4, 0, 3, 6, 7, 5, 8, 2

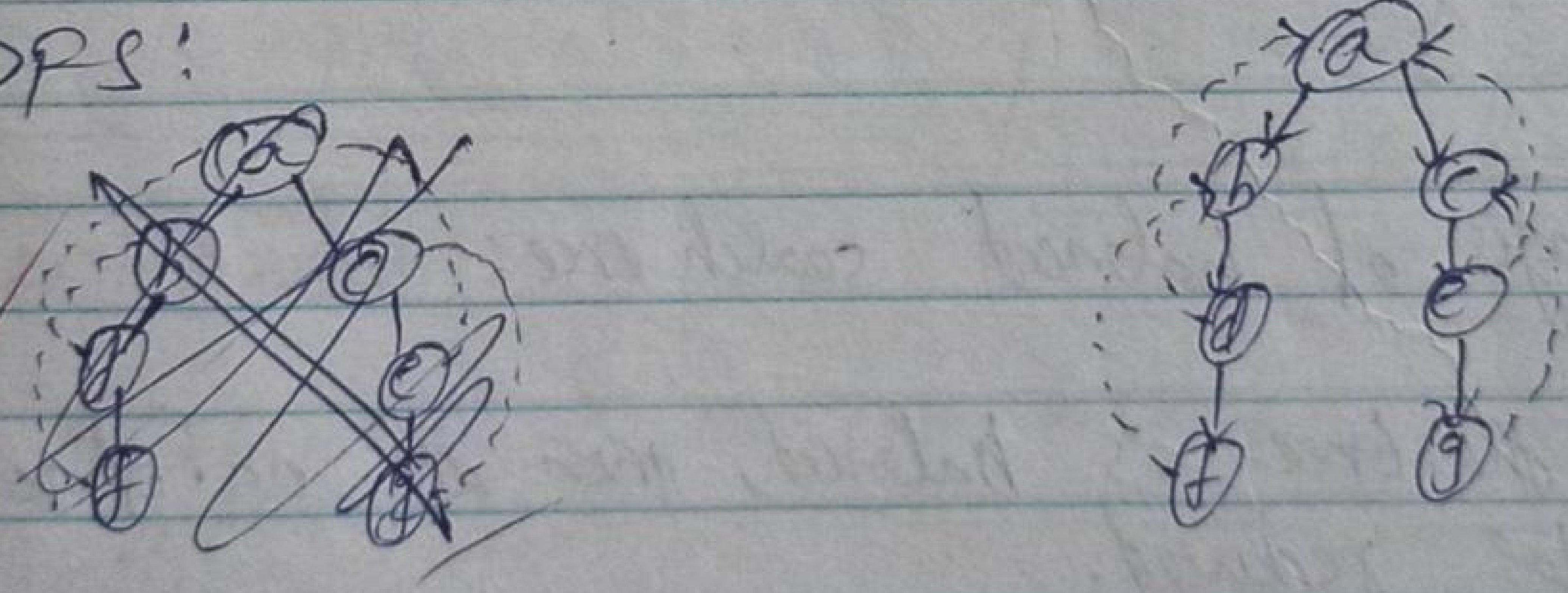


5) Three Major variations of Decrease and Conquer approach:

Eg: To find a^n,

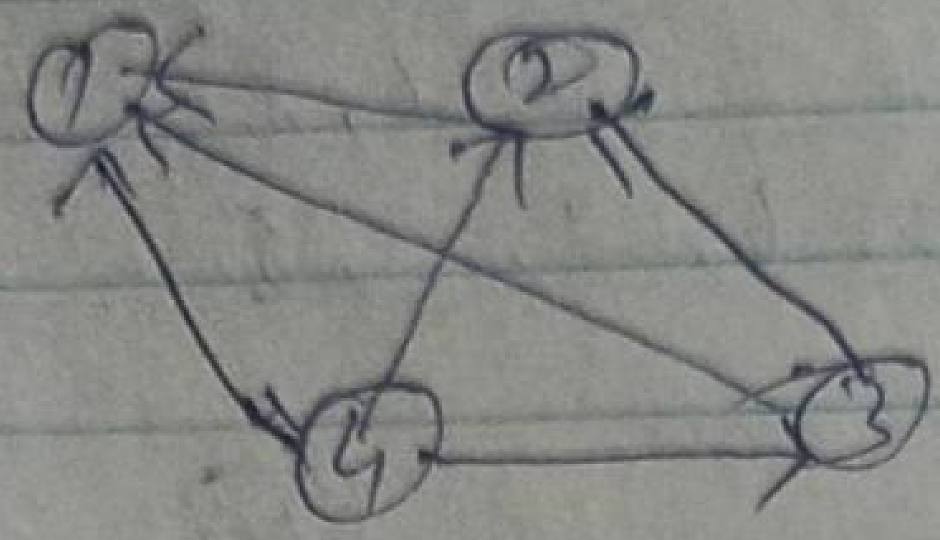
* Decrease by constant – $\begin{cases} f(n-1)a, & n>1 \\ a, & n=1 \end{cases}$
* Decrease by constant factor – $\begin{cases} (a^{n/2})^2, & n \text{ is even} \\ (a^{(n-1/2)})^2, & n \text{ is odd} \\ a & n=1 \end{cases}$
* Decrease by variable size of decrease – $\begin{cases} a^{n/2} * a^{n/2}, & n>1 \\ a & n=1 \end{cases}$

6) DFS:



DFS order: abdfceg

7)



Ans:
Topological sort order: 4, 3, 2, 1

8) Alg. Uniqueness using presorting

Alg uniqueness (A[0..n-1])
{
for i = 0 to n-1
{ if (A[i] == A[i+1])
return FALSE
}
return TRUE
}

// if False -> not unique
if True -> unique

9) Advantages of balanced search tree:

* If the tree is balanced, then the no. of searches will be reduced.
* we can skip the any one side (left or right) based on root while searching, if the tree is not balanced, it may take time to search in tree.

1) Master Theorem:

The theorem states that, the problem belongs to the following order in the case of $aT(n/b) + c$ where $c = n^d$

$O(n^d)$, if $a < b^d$

$O(n^d \log n)$, if $a = b^d$

$O(n \log_b a)$ if $a > b^d$

10) $\underset{a_1}{21}\,\underset{a_0}{01} * \underset{b_1}{11}\,\underset{b_0}{30}$

$c_2 = a_1 * b_1$

$= \underset{a_1}{2}\underset{a_0}{1} * \underset{b_1}{1}\underset{b_0}{1}$

$c_2 = 231$

$c_0 = 30$

$c_1 = 792$

$c = 2310000 + 99200 + 30 + 330 - 231 - 30$

~~c =~~

$c = 2374130$

$21 * 11$

$c_2 = 2$

$c_0 = 1$

$c_1 = 3 * 2 - 3 = 2$

$c = 200 + 1 + 30$

$c = 231$

$1 * 30 = 330$

~~01 * 30~~

$c_2 = 0$

$c_0 = 0$

$c_1 = 3$

$c = 30$

$32 * 31$

$c_2 = 9$

$c_0 = 2$

$c_1 = 5 * 4 - 11 = 9$

$c = 900 + 90 + 2 = 992$

3) Alg height(root r)

{

if(r==null)

return 0;

else

return(height(r->left) - height(r->right) + 1)

}