UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

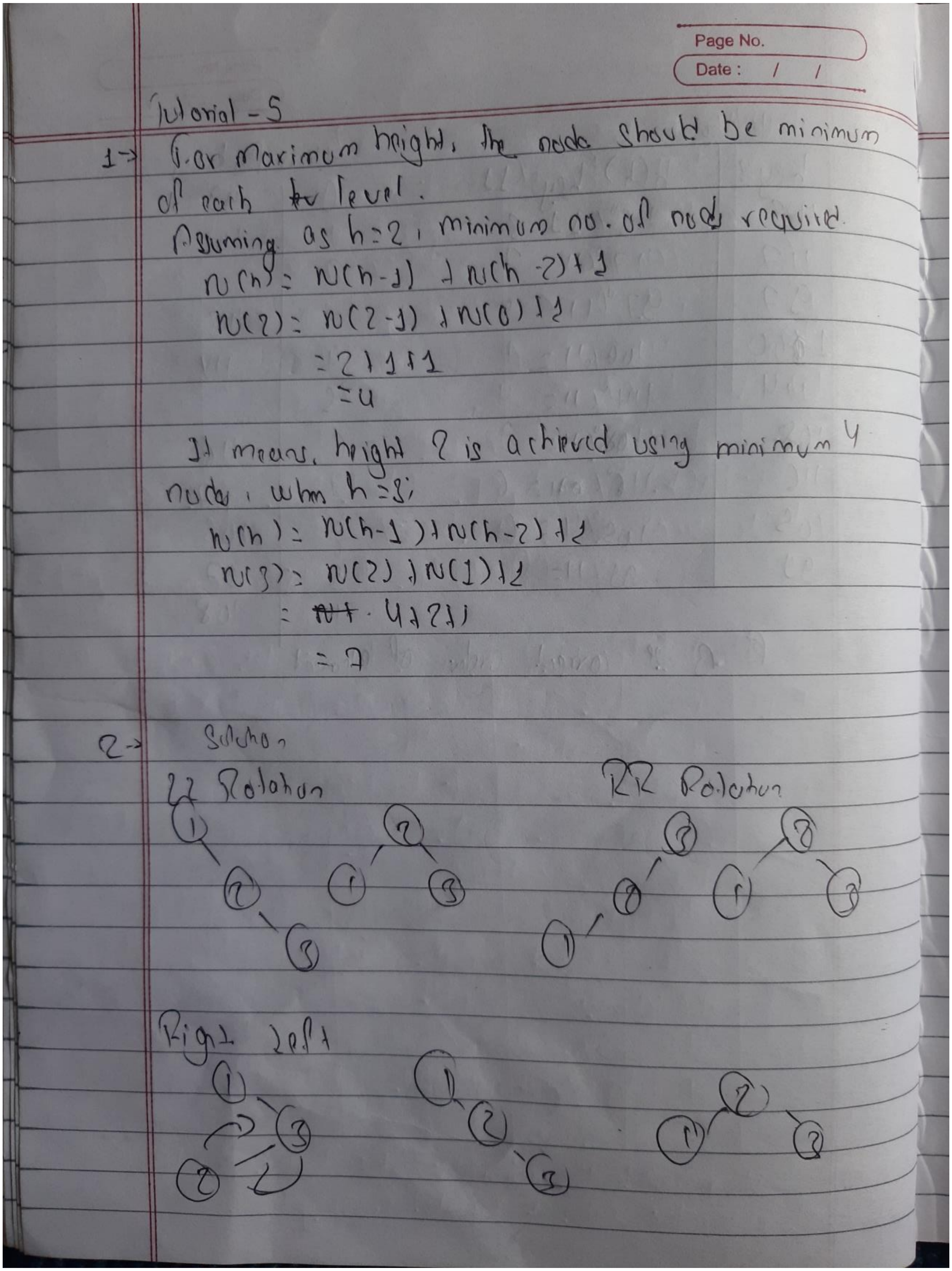
iAcademy
Quality Education Improving Lives

HERALD COLLEGE
KATHMANDU

Academic Year	Module	Assessment Number	Assessment Type
S20	Introductory Data Structures and Algorithms (DipIT02)	A1	Assignment Submission

[Assignment Submission]

Student Id	: [NP03A190299]
Student Name	: [Yogesh Shrestha]
Section	: [DC8]
Module Leader	: [Mr. Prakash Gautam]
Submitted on	: 06-03-2020



Page No.
Date : / /

Tutorial - 5

1-> For Maximum height, the node should be minimum of each ~~to~~ level.

Assuming as h=2, minimum no. of node required

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

$n(2) = n(2-1) + n(0) + 1$

$= 2 + 1 + 1$

$= 4$

It means, height 2 is achieved using minimum 4 node, when h=3;

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

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

$=$ ~~n+~~ $4 + 2 + 1$

$= 7$

2-> Solution

LL Rotation

1
2
3
2
1
3

RR Rotation

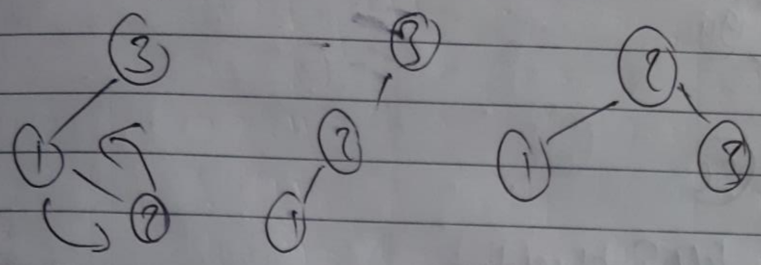
3
2
1
2
1
3

Right left

1
3
2
1
2
3
2
1
3

Page No.
Date : / /

left Right



3 → . The worst case possible height of AVL tree with n nodes is $1.44 \times \log h$.

4 → ~~O~~$(\log n)$ Sol^n

Worst case running time (W(RT) for balanced BST (AVL) tree with N nodes is $O(\log N)$

we have, $N = 2 \times 2^n$

So. $W(RT = O(\log(n \times 2^n)))$

Since $\log(ab) = \log a + \log b$,

$W(RT = O(\log n + \log 2^n)$

$W(RT = O(\log^n + n)$

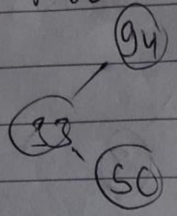
Since $O(\log n) < (n)$ we have

$W(RT = O(n)$

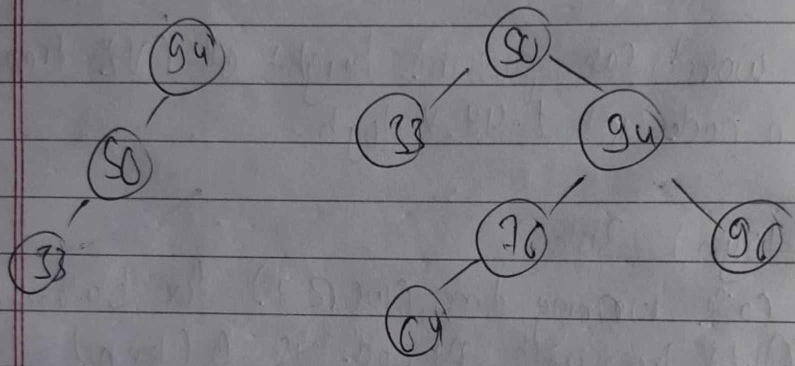
5 → . The maximum number of nodes of AVL trees possible with height h is $2^{h+1} - 1$.

Page No.
Date : / /

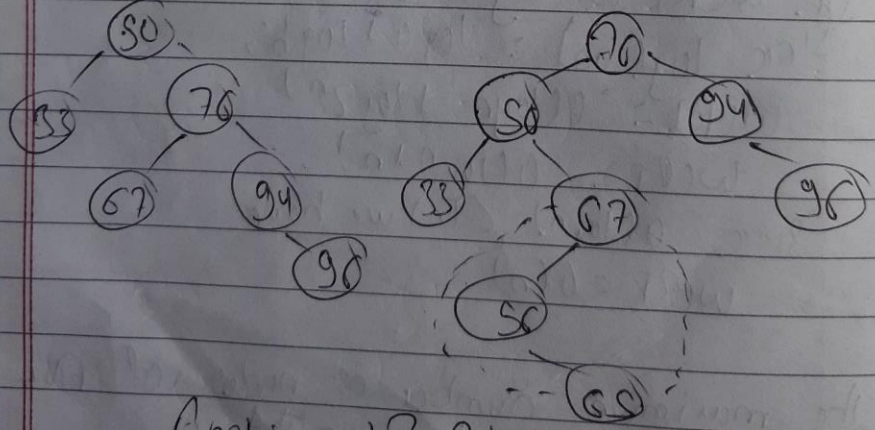
6 -> AVL tree



Applying left Right rotation

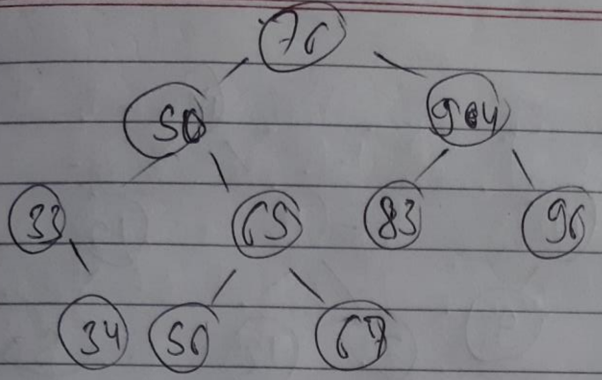


Applying Right left rotation

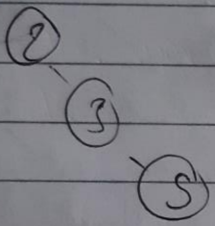


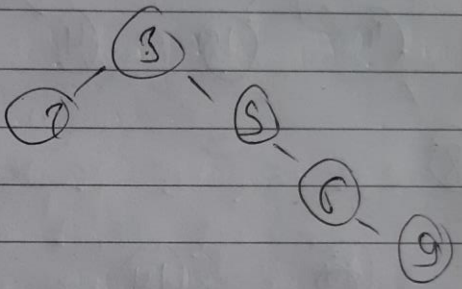
Applying LR Rotation

Page No.
Date : / /



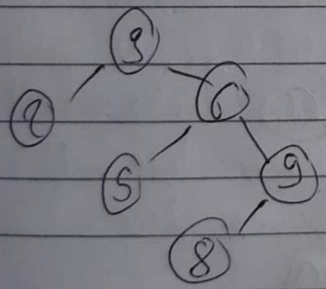
7.)

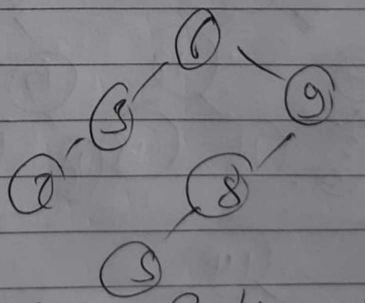




Applying left rotation

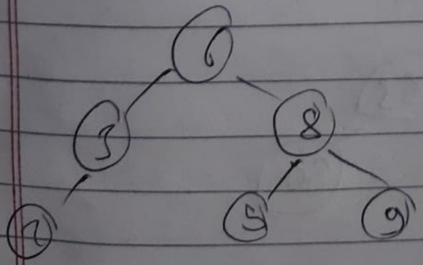
Applying left rotation





Applying left rotation

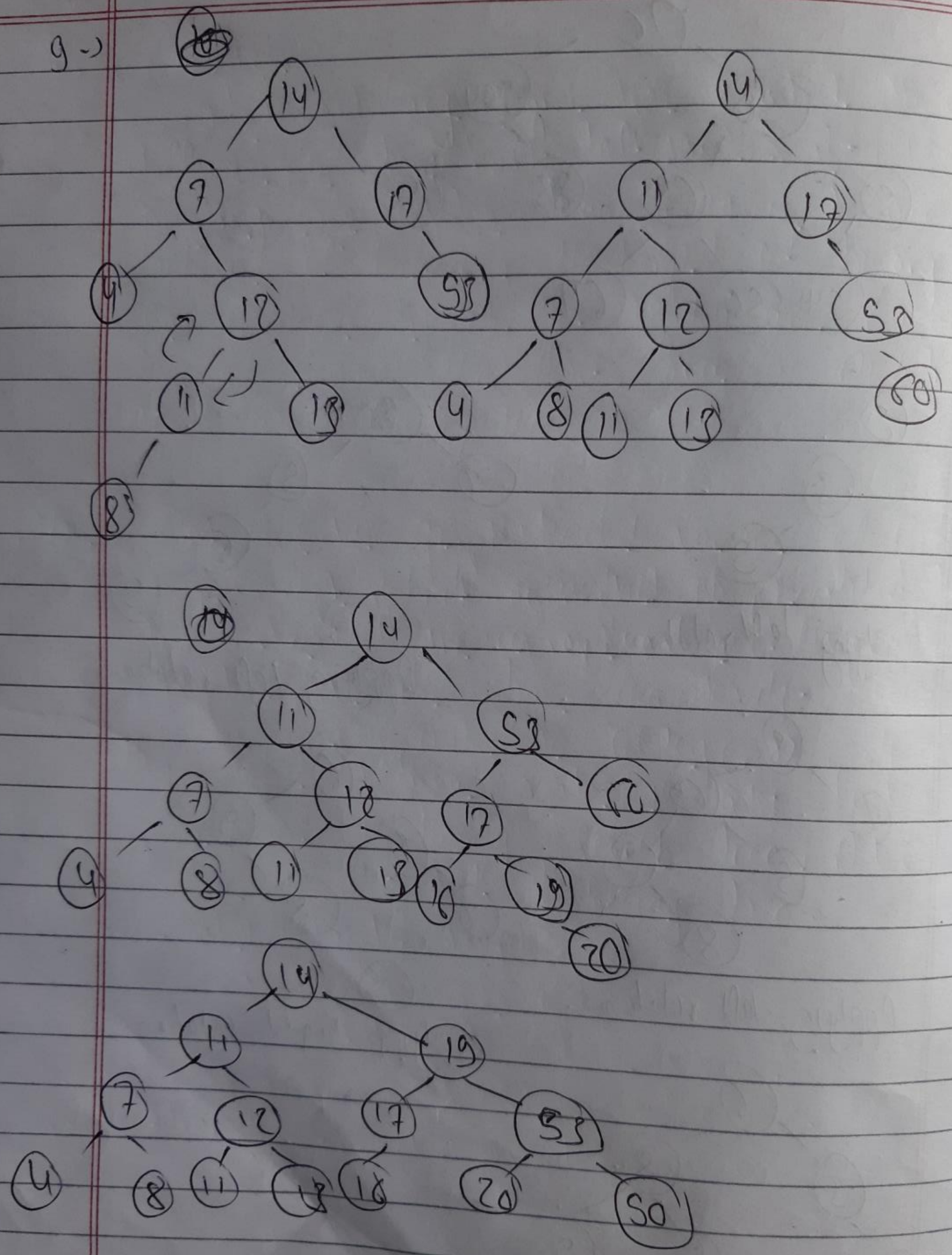
Applying Right rotation



Page No.

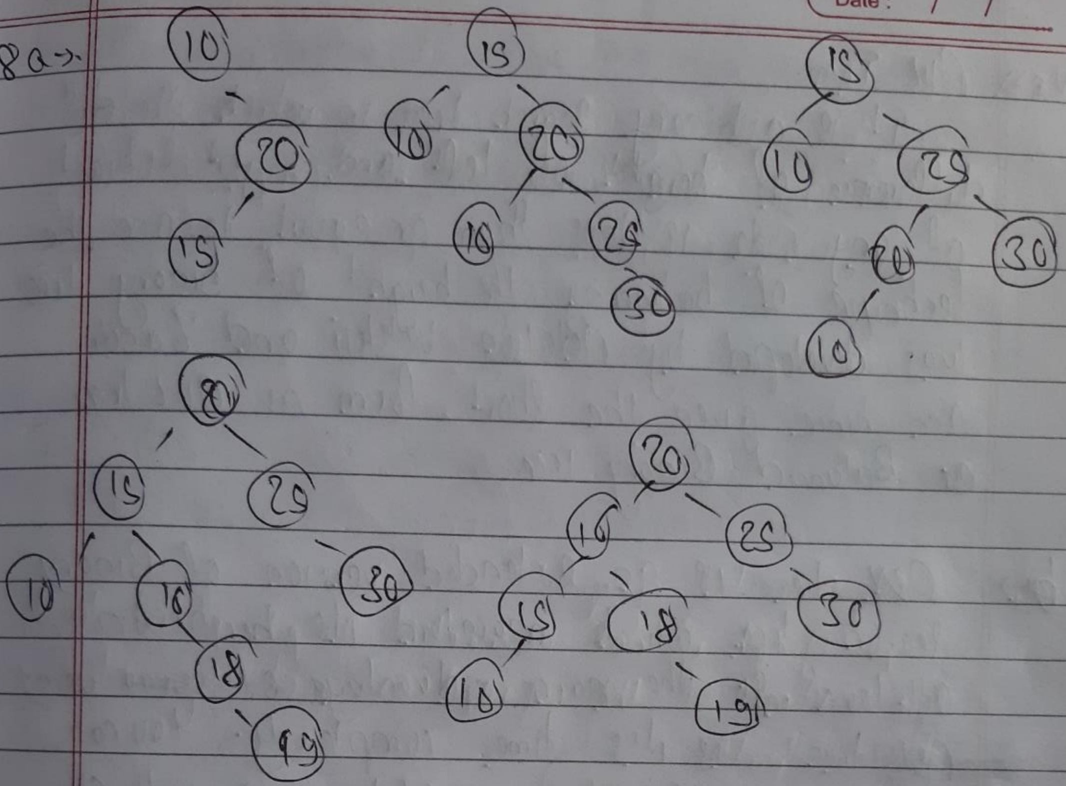
Date : / /

9.)

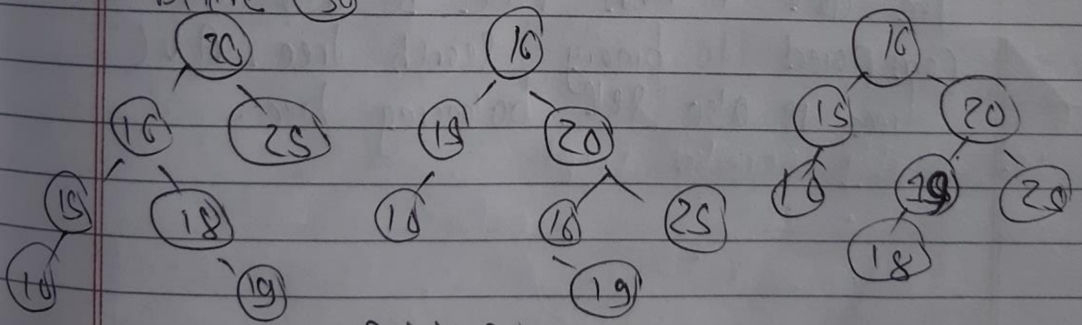


Page No.
Date : / /

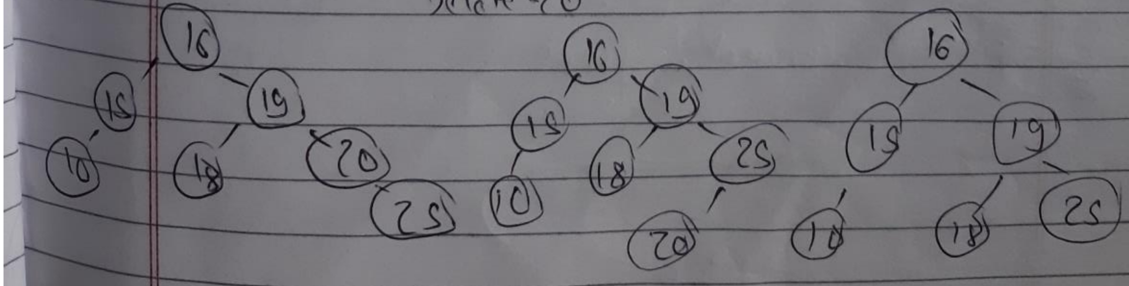
8a->

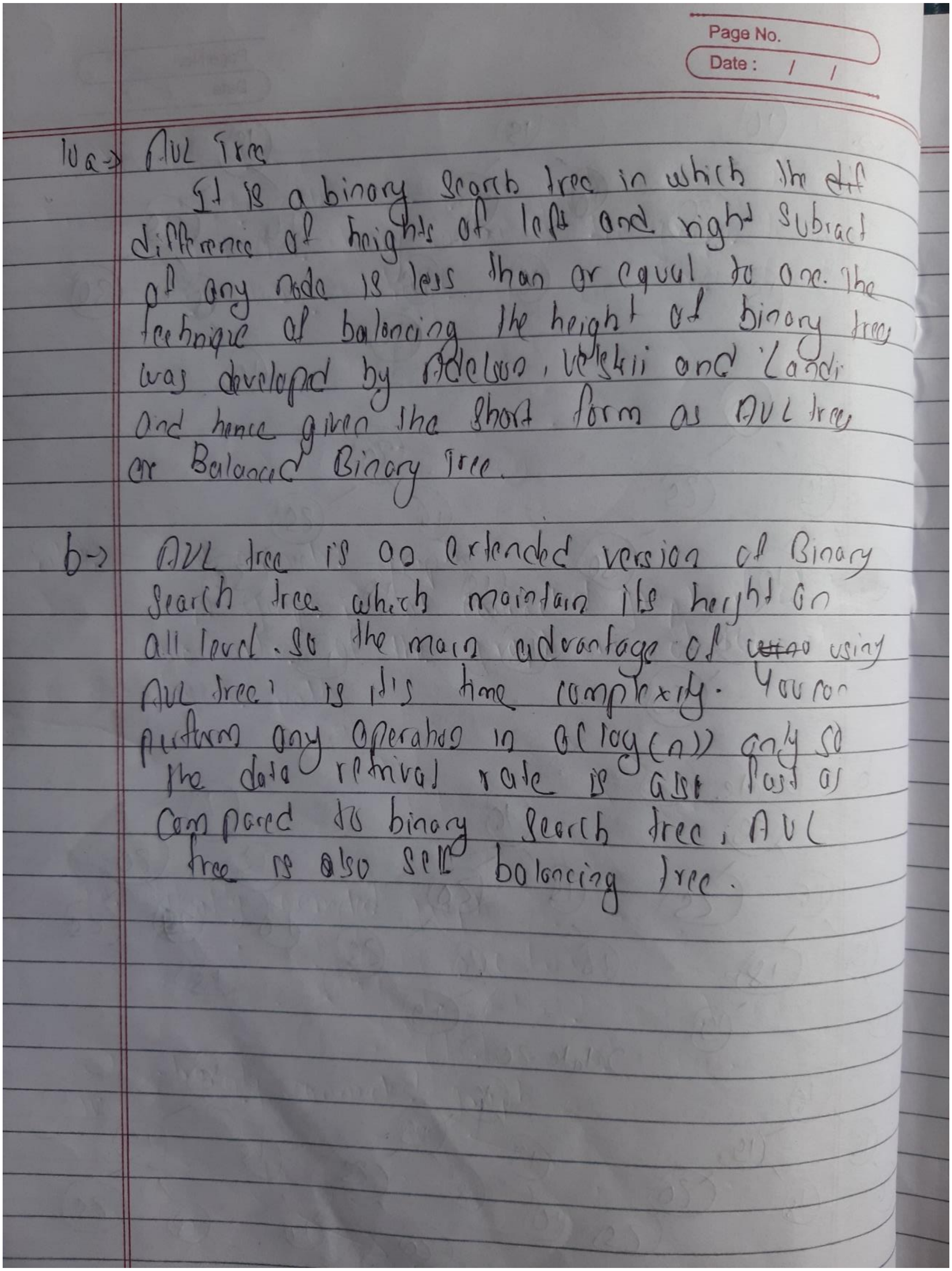


b Delete 30



Delete -20





Page No.
Date : / /

10a-> AVL Tree

It is a binary Search tree in which the ~~dif~~ difference of heights of left and right Subract of any node is less than or equal to one. The technique of balancing the height of binary trees was developed by Adelson, Velskii and Landi And hence given the short form as AVL tree or Balanced Binary Tree.

b-> AVL tree is an extended version of Binary Search tree which maintain its height on all level. So the main advantage of ~~using~~ using AVL tree is it's time complexity. You can perform any operation in O(log(n)) only so the data retrival rate is also fast as compared to binary Search tree. AVL tree is also self balancing tree.