FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC03A3/CSC3A10 COMPUTER SCIENCE 3A
CAMPUS	AUCKLAND PARK CAMPUS (APK)
ASSESSMENT	AVL TREE EXAMPLES **MEMO**

DATE: 2023-05-10 **SESSION:** Practice

ASSESOR(S): PROF D.T. VAN DER HAAR
PROF H. VADAPALLI

DURATION: 36 MINUTES **MARKS:** 30

Please read the following instructions carefully:

1. Answer **all** the questions
2. Write *cleanly* and *legibly*.
3. You may use a non-programmable calculator to answer the questions.
4. This paper consists of 13 pages.

QUESTION 1

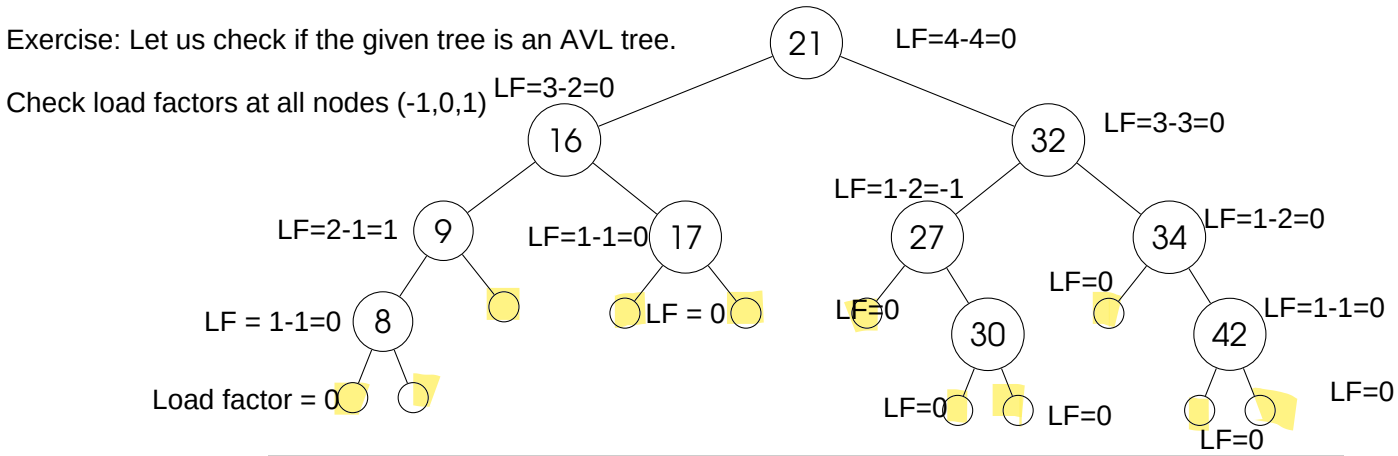
(a) Consider the following AVL tree provided below. Draw the AVL tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced. Removal operations should follow from the tree that resulted from the insertion operations. (15)

1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order)

   17, 29, 14, 26, 24

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order)

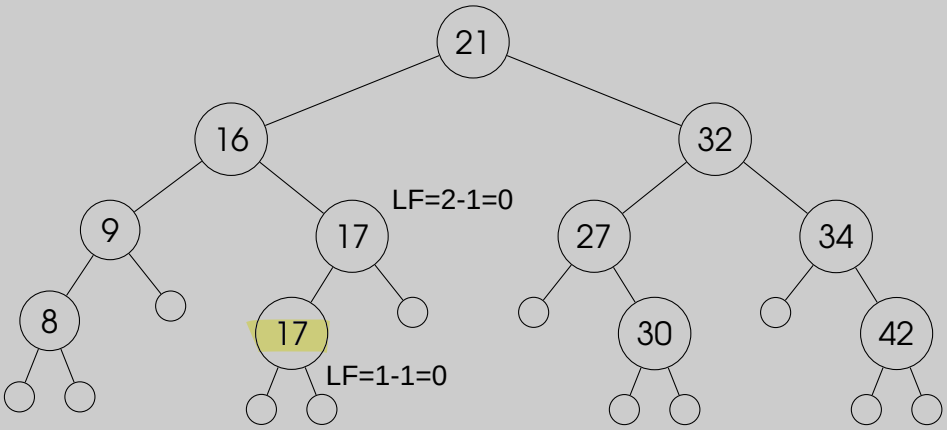
   17, 32, 14, 24



Solution:

Insert 17

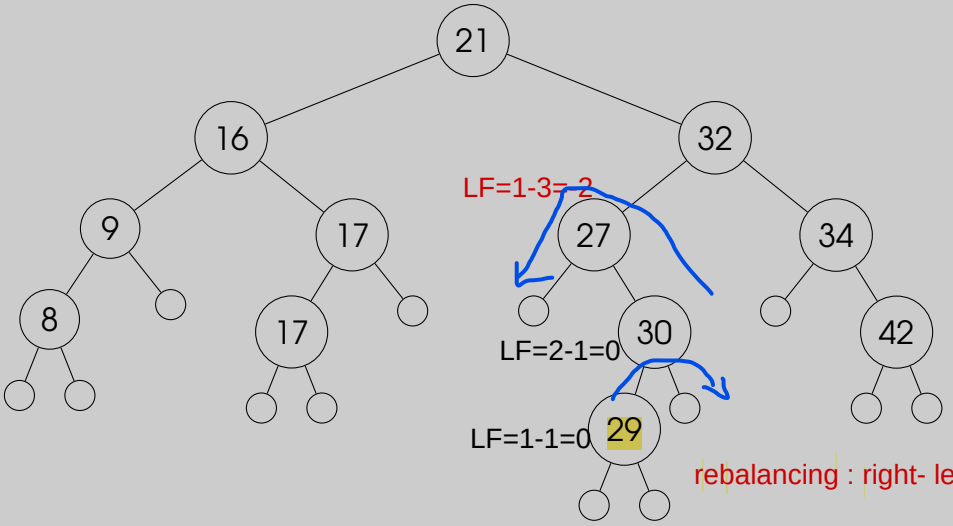
(1 mark):



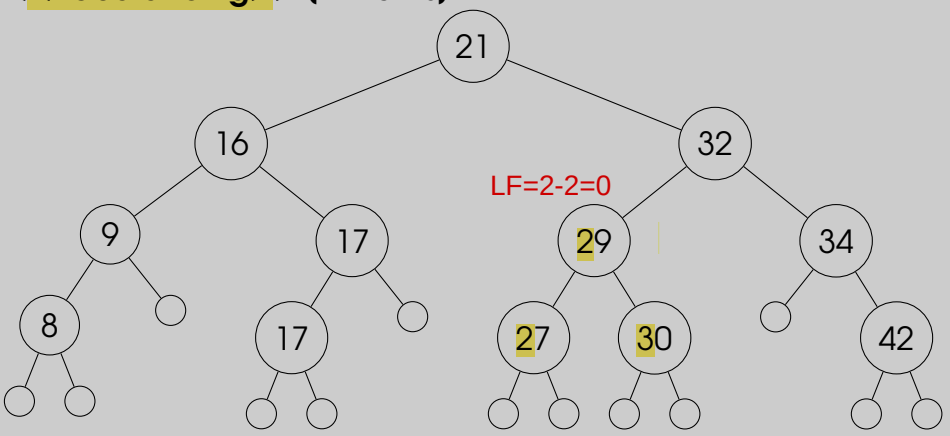
Insert 29

Check load factors are in (-1,0,1)

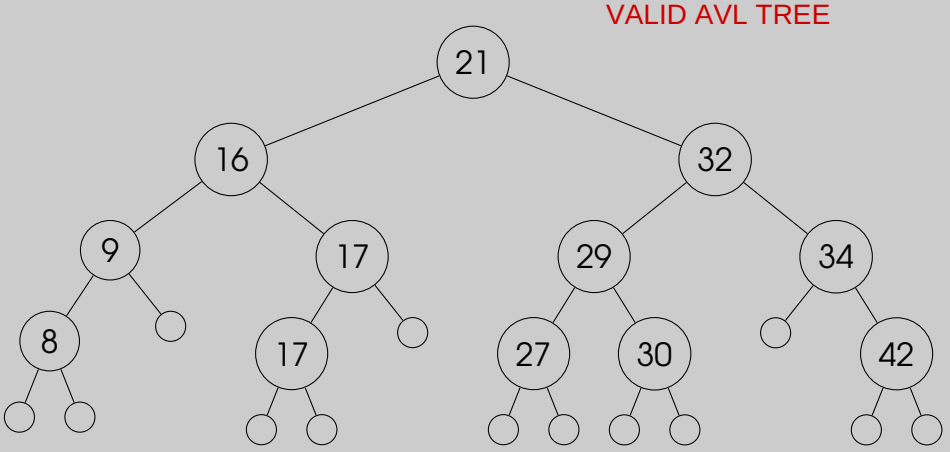
(1 mark):



<<**Rebalancing**>> **(2 marks):**

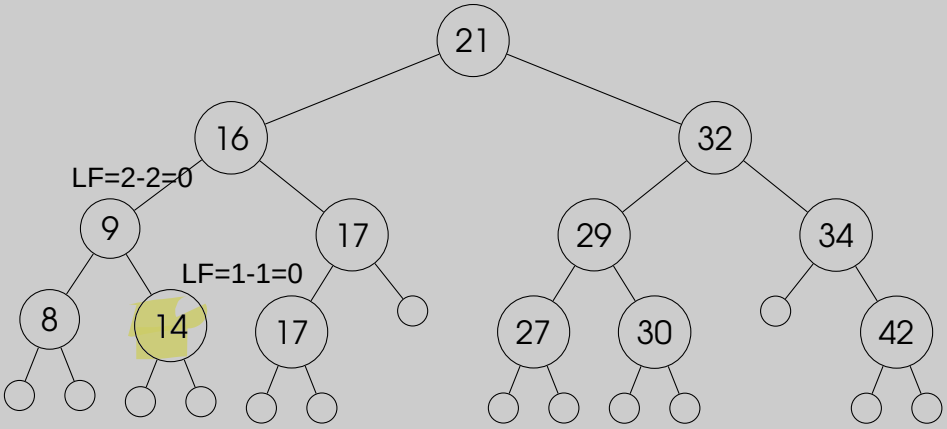


VALID AVL TREE



Insert 14

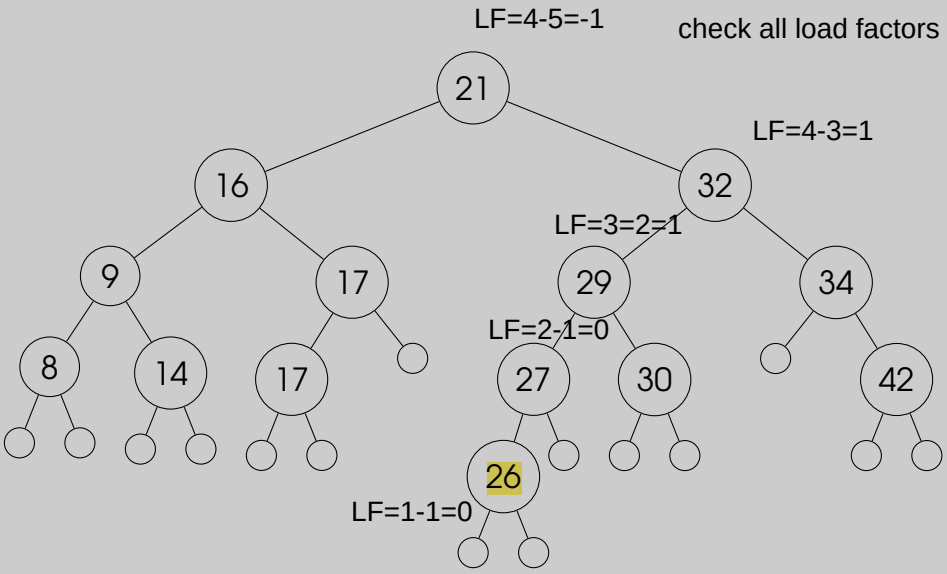
(1 mark):



Load factors requirements met

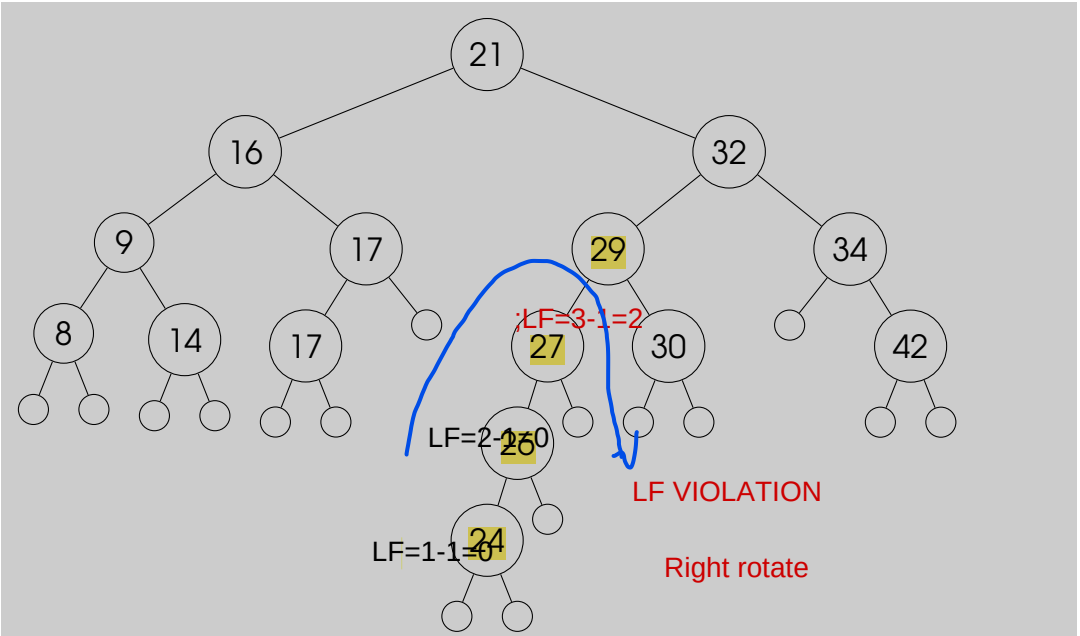
Insert 26

(1 mark):

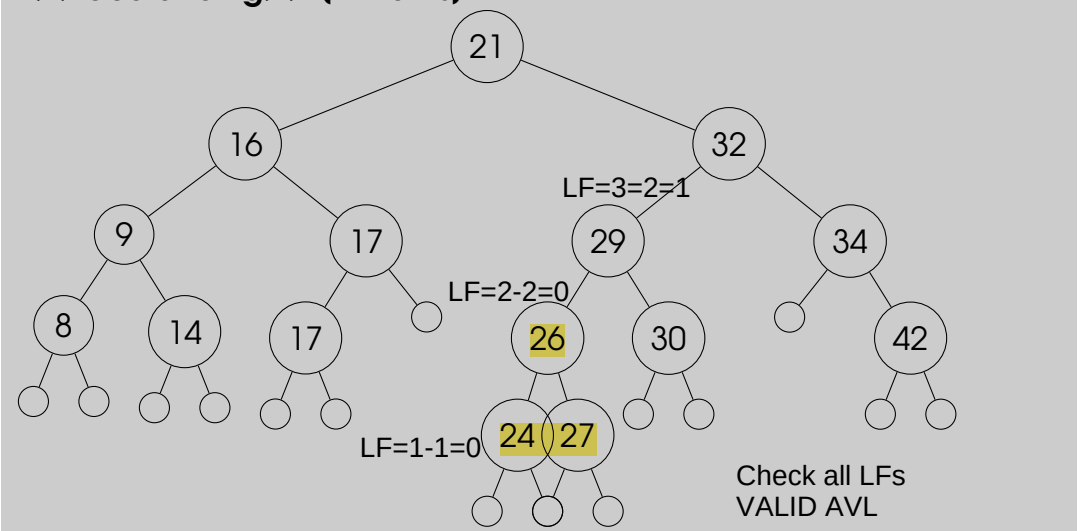


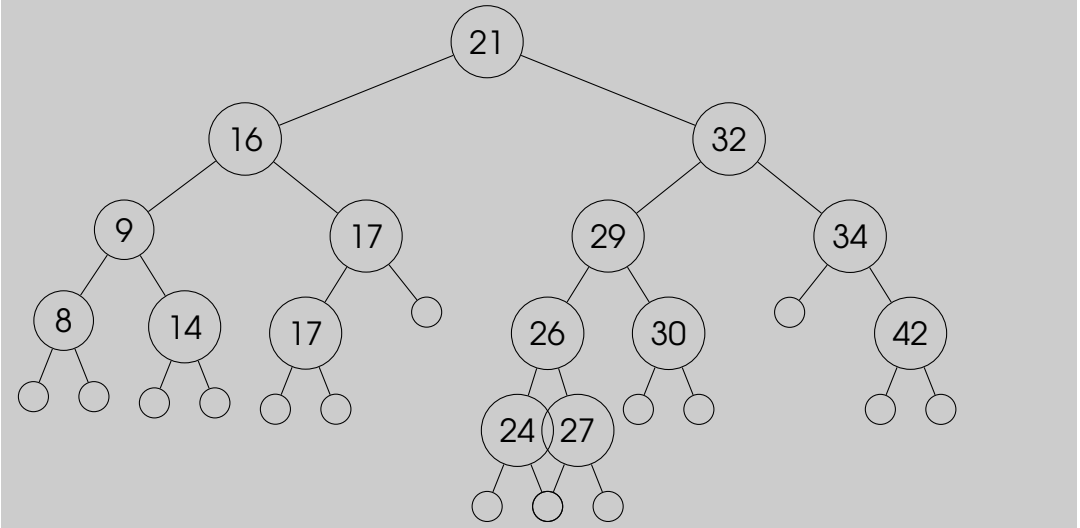
Insert 24

(1 mark):



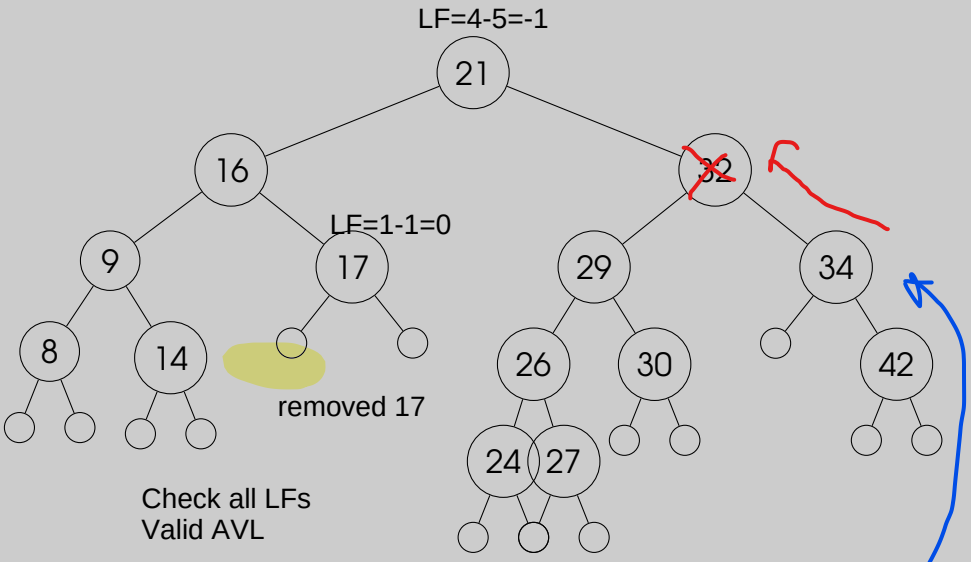
<<**Rebalancing**>> **(2 marks):**





Remove 17

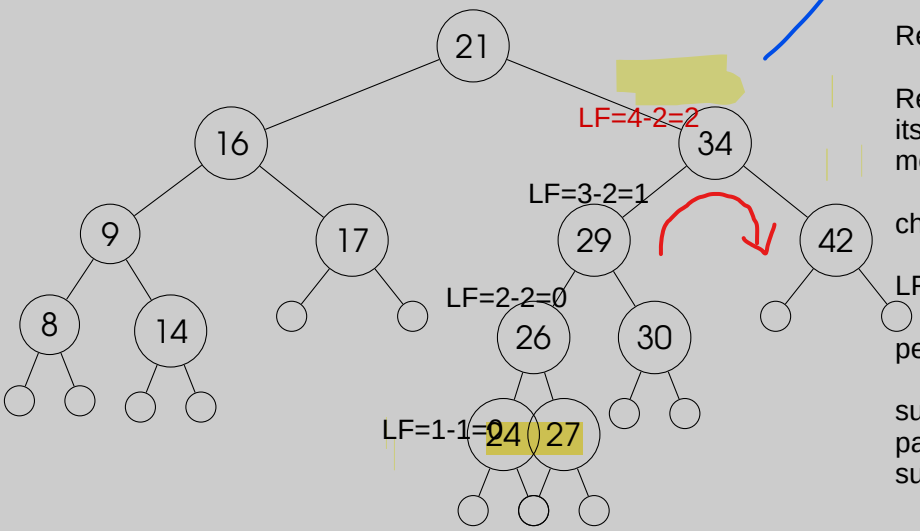
(1 mark):



Check all LFs
Valid AVL

Remove 32

(1 mark):



Removed node 32,

Replace deleted node with its inorder successor; move 34 to new empty slot

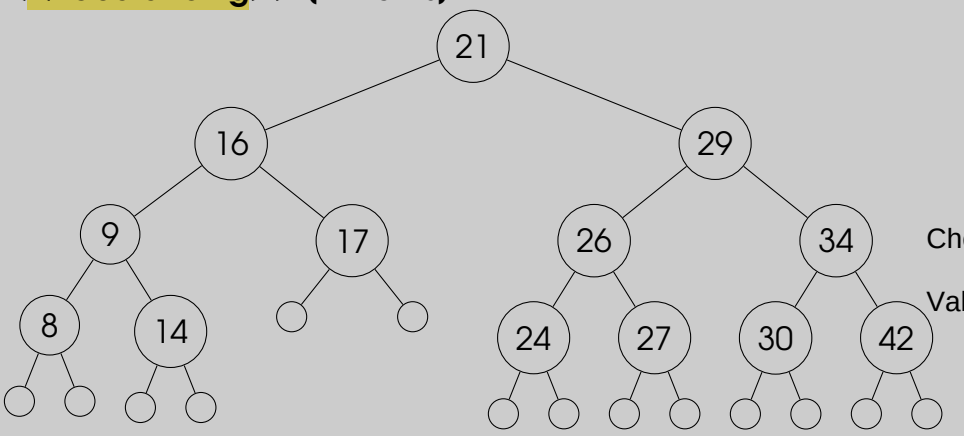
check LFs

LF 2 at node 34

perform right rotation

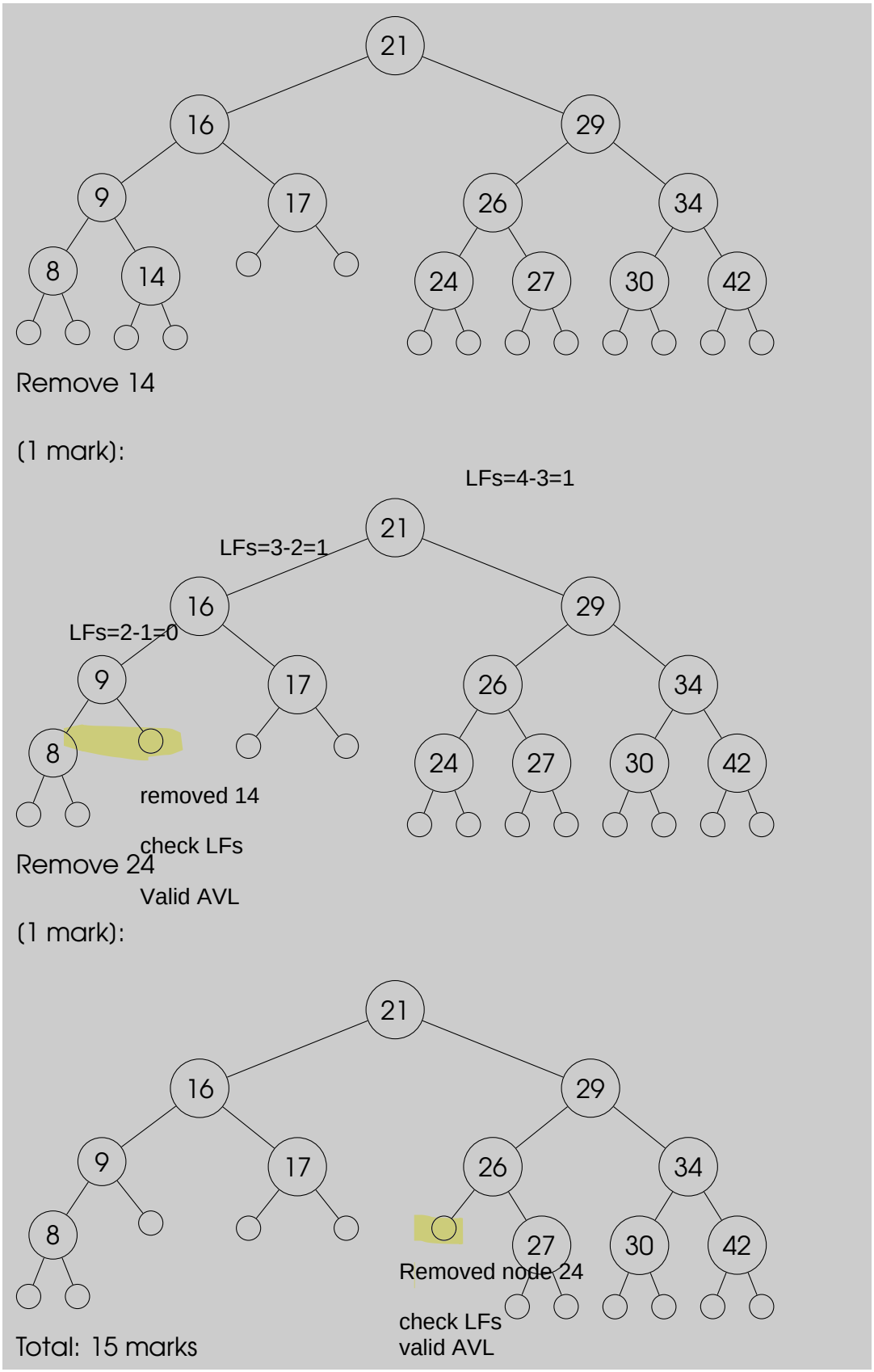
subtree with node 30 as parent moves as the left subtree for node 34

<<**Rebalancing**>> **(2 marks):**



Check LFs

Valid AVL



(b) Consider the following AVL tree provided below. Draw the AVL tree state after each of the following operations. If the tree is rebalanced (15)

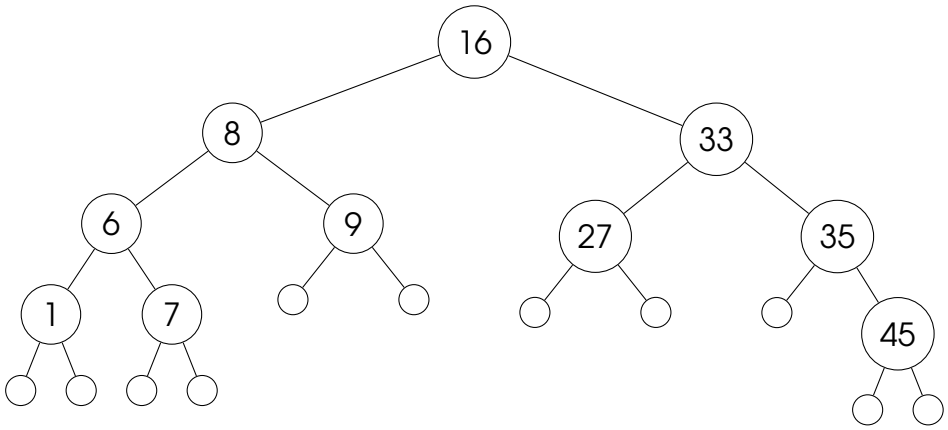
draw the state before and after it being balanced. Removal operations should follow from the tree that resulted from the insertion operations.

1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order)

   23, 23, 20, 29, 8

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order)

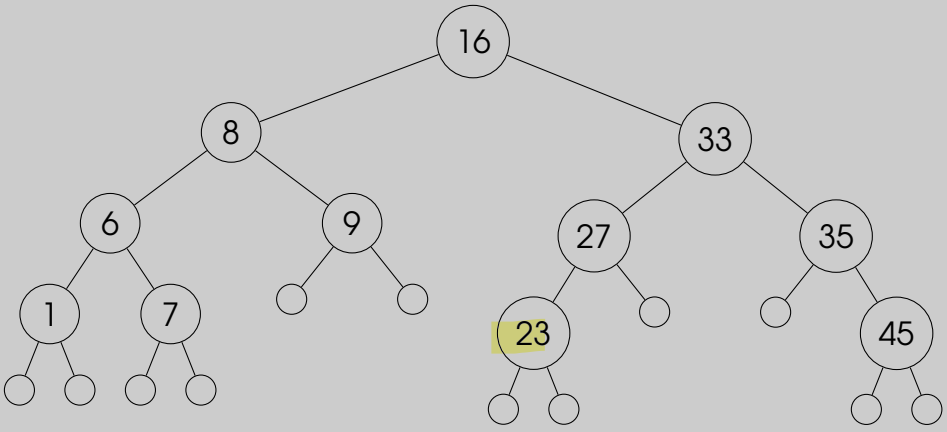
   1, 33, 9, 27



Solution:

Insert 23

(1 mark):

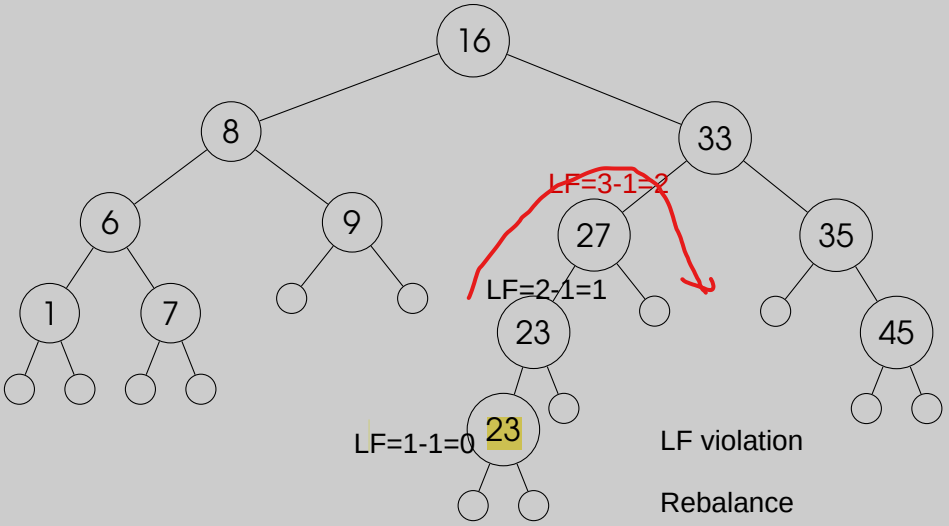


Insert 23

Check LFs

(1 mark):

Valid AVL

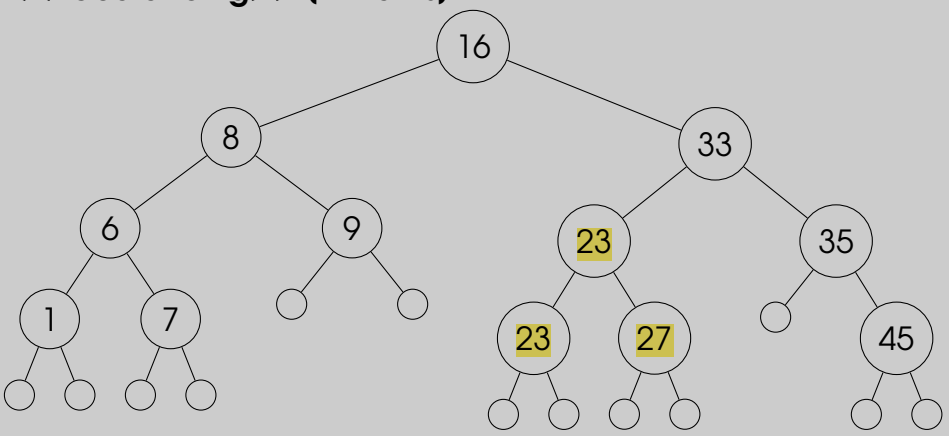


LF violation

Rebalance

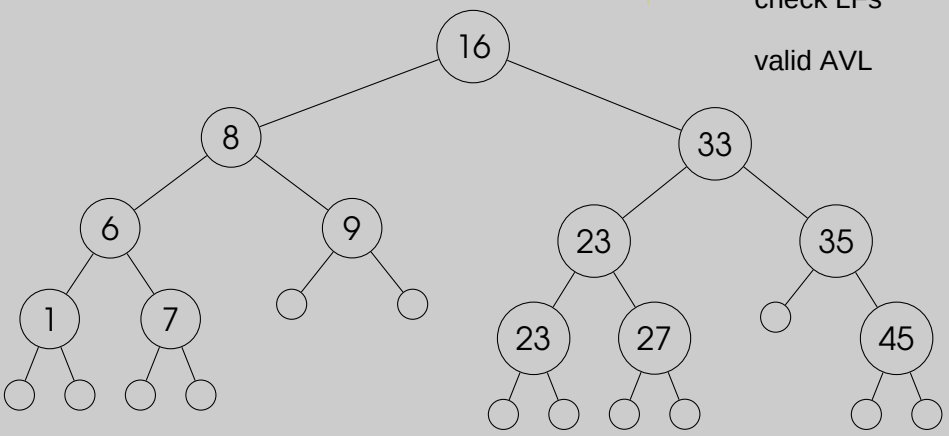
perform right rotation

<<**Rebalancing**>> **(2 marks):**



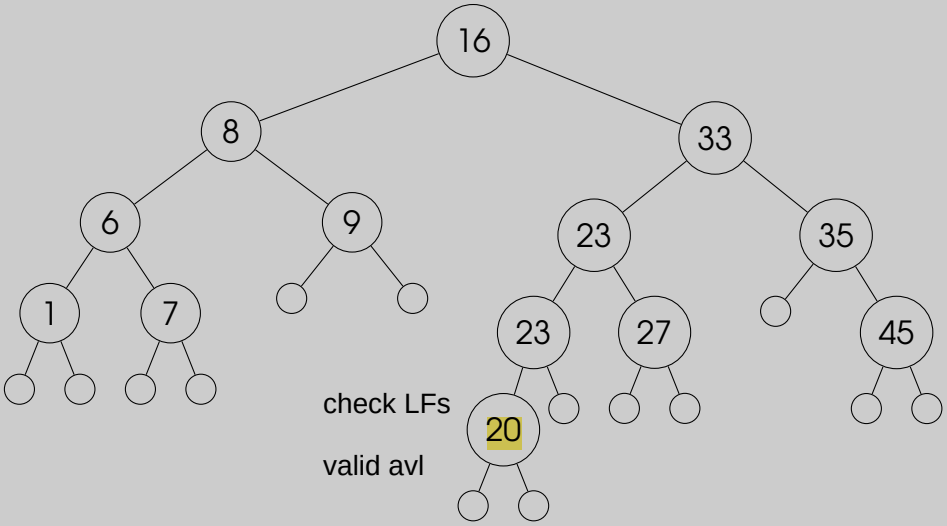
check LFs

valid AVL



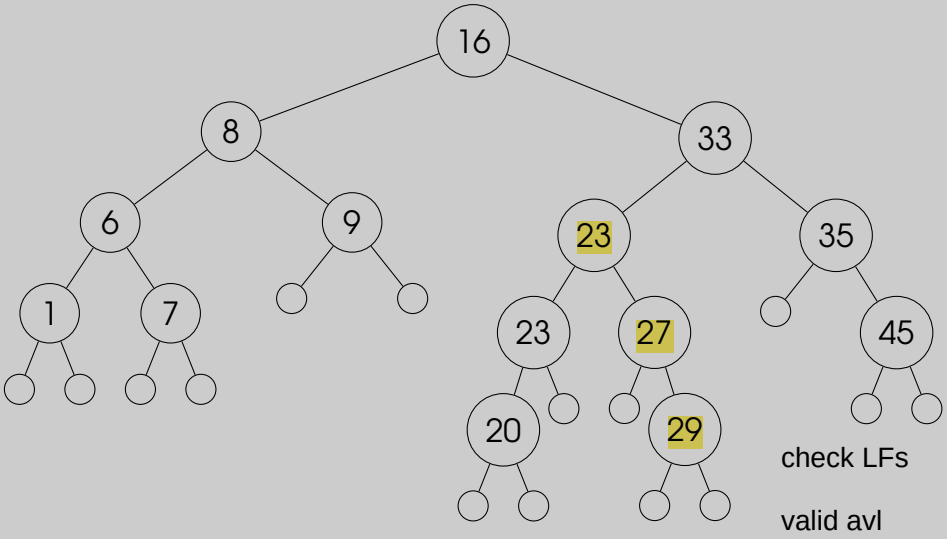
Insert 20

(1 mark):



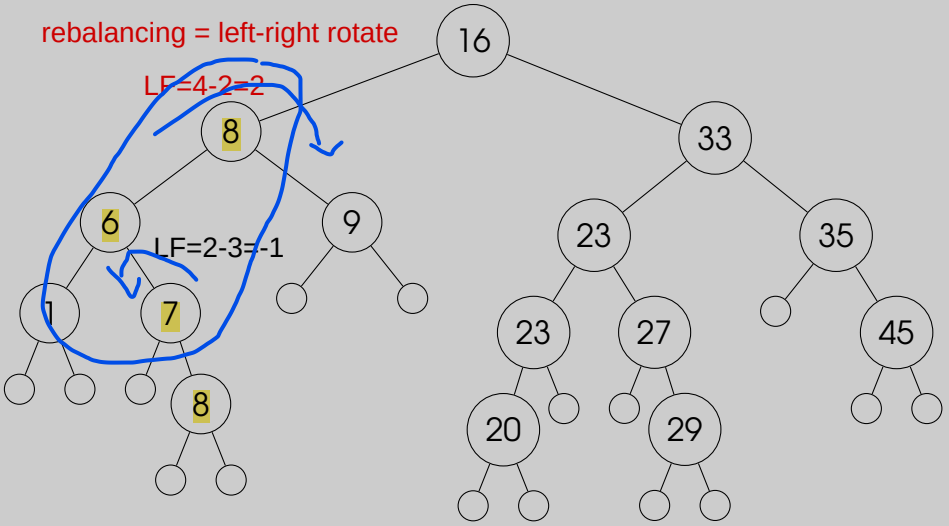
Insert 29

(1 mark):

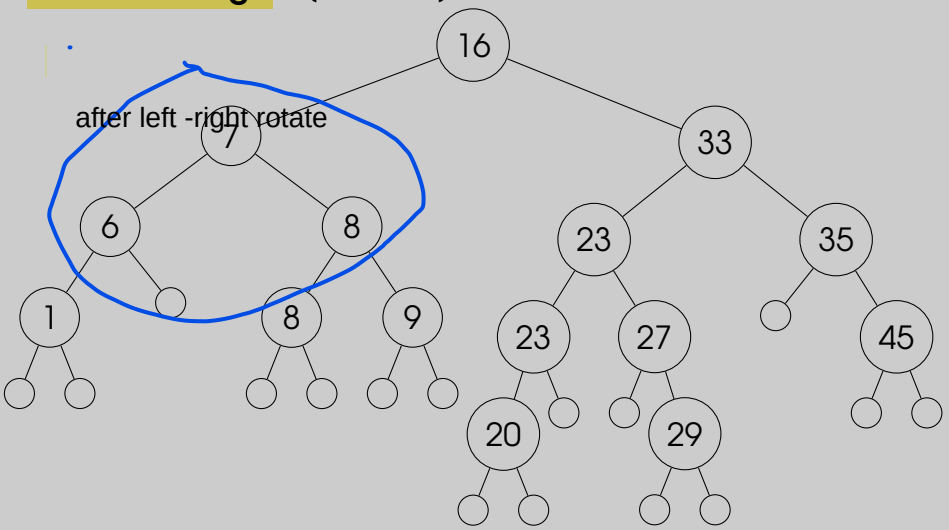


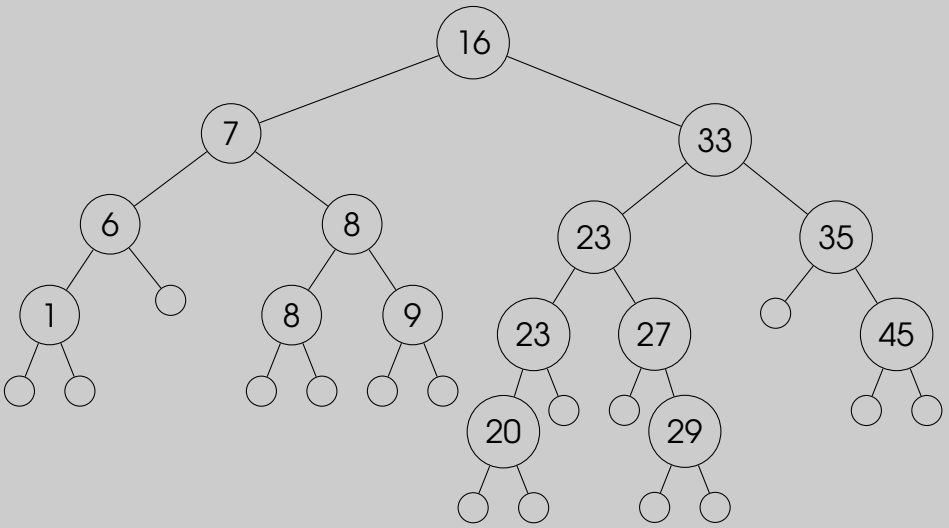
Insert 8

(1 mark):



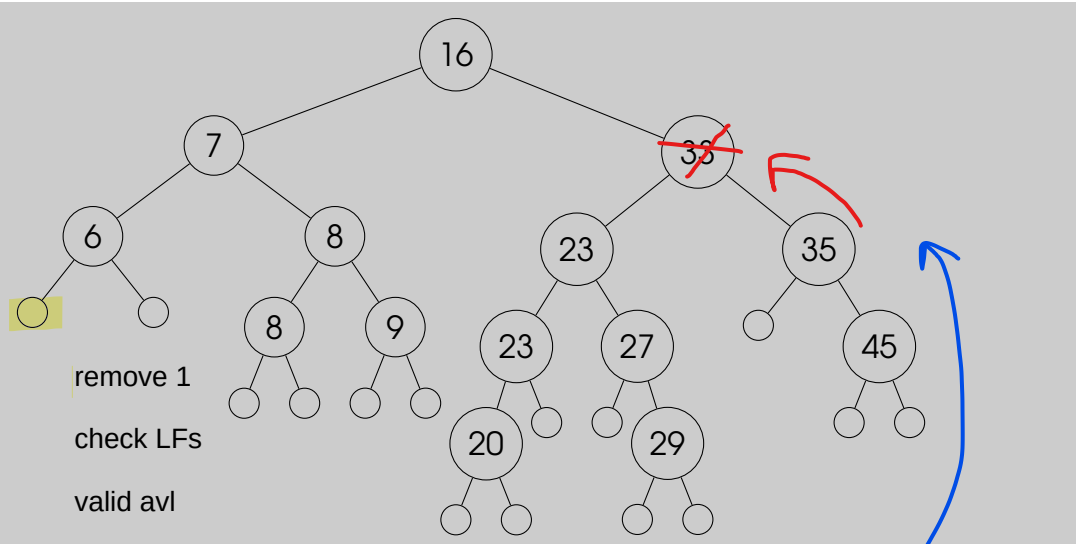
<<**Rebalancing**>> **(2 marks):**





Remove 1

(1 mark):



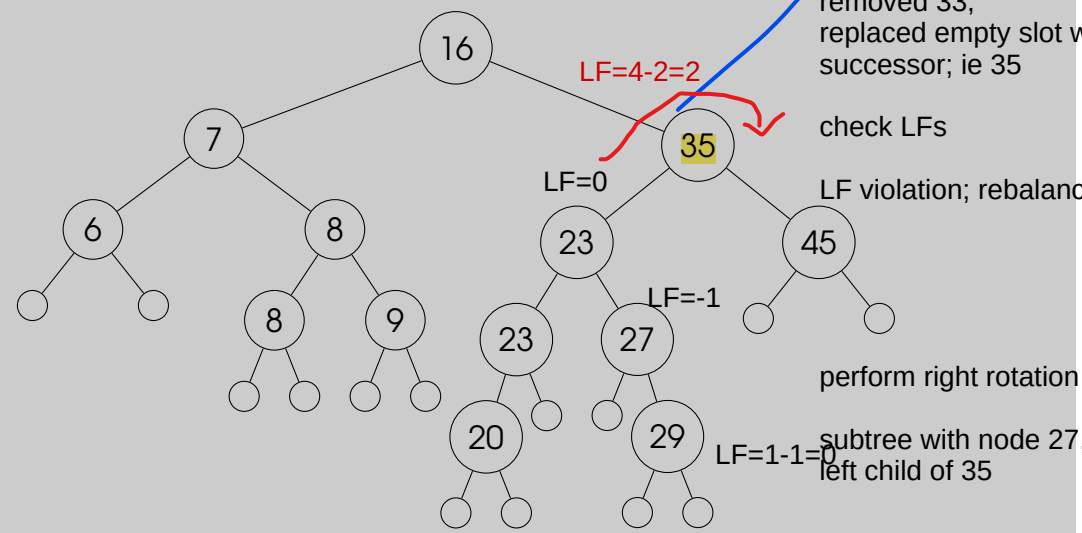
remove 1

check LFs

valid avl

Remove 33

(1 mark):



removed 33,
replaced empty slot with in-order successor; ie 35

LF=4-2=2

check LFs

LF=0

LF violation; rebalance

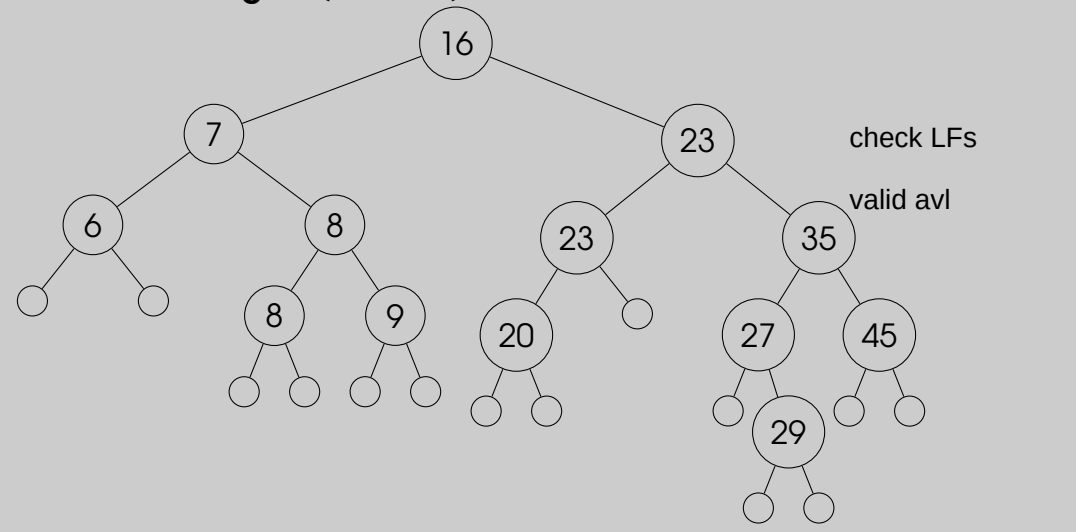
LF=-1

perform right rotation

LF=1-1=0

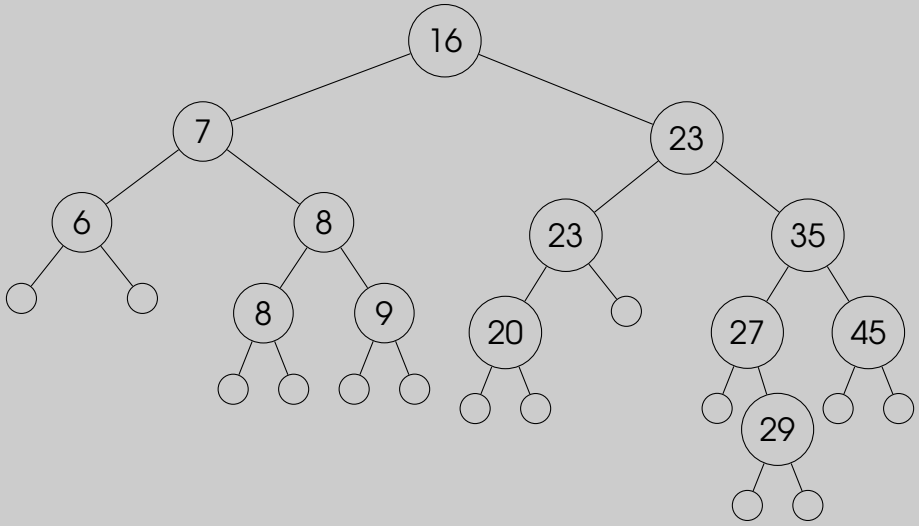
subtree with node 27, moves as left child of 35

<<**Rebalancing**>> **(2 marks):**



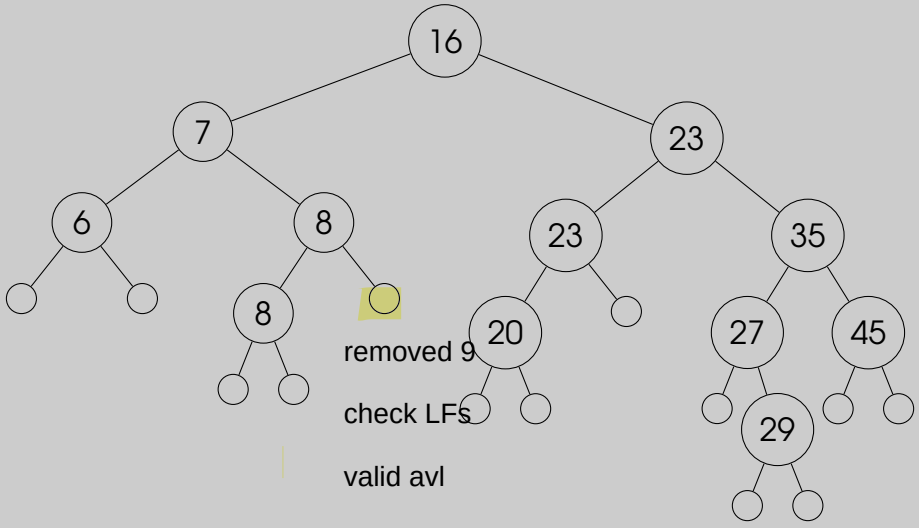
check LFs

valid avl



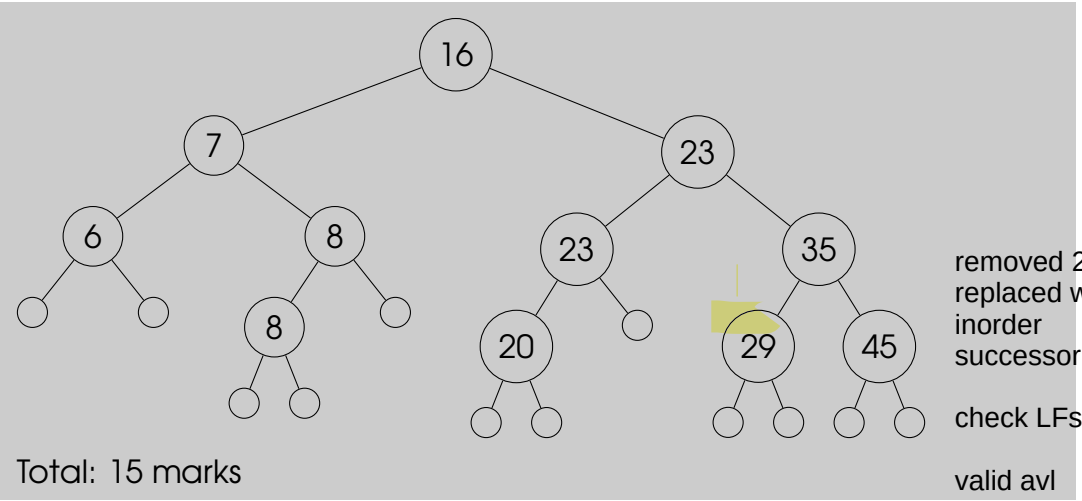
Remove 9

(1 mark):



Remove 27

(1 mark):



Total: 30

— End of paper —