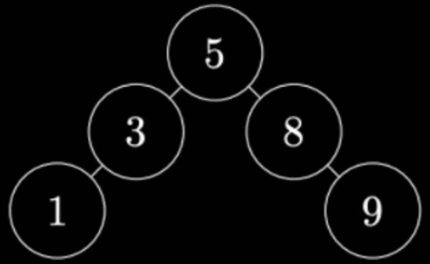
Practice set: AVL TREES

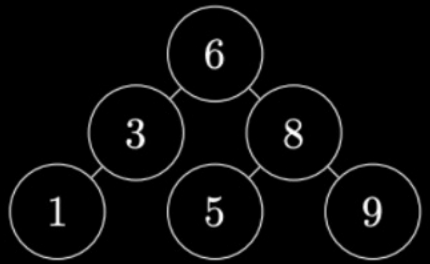
Attempt all the questions before referring to the answers.

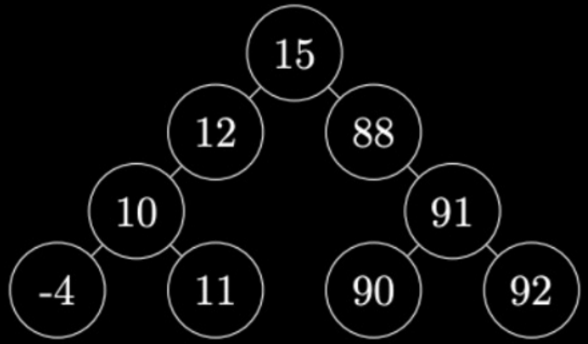
If the answers are not provided in the slides, please write your answer in the comments section to get it verified with others.

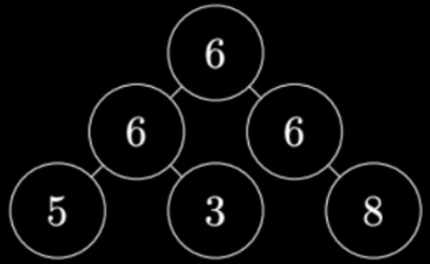
Question: 1

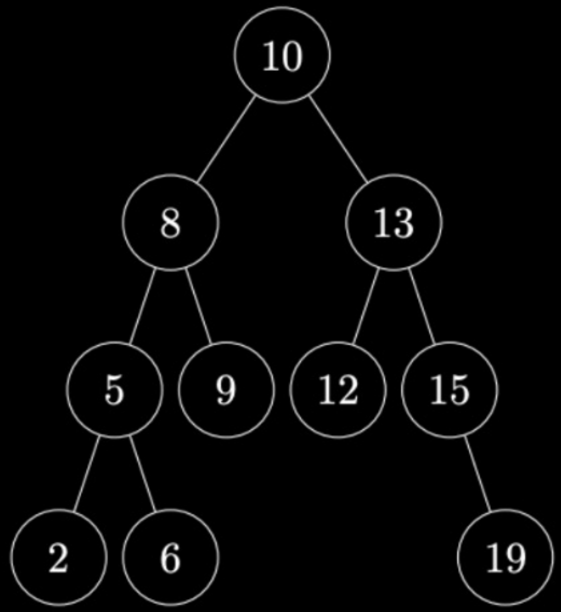
Are the following trees valid AVL trees?

(a)


(b)


(c)


(d)


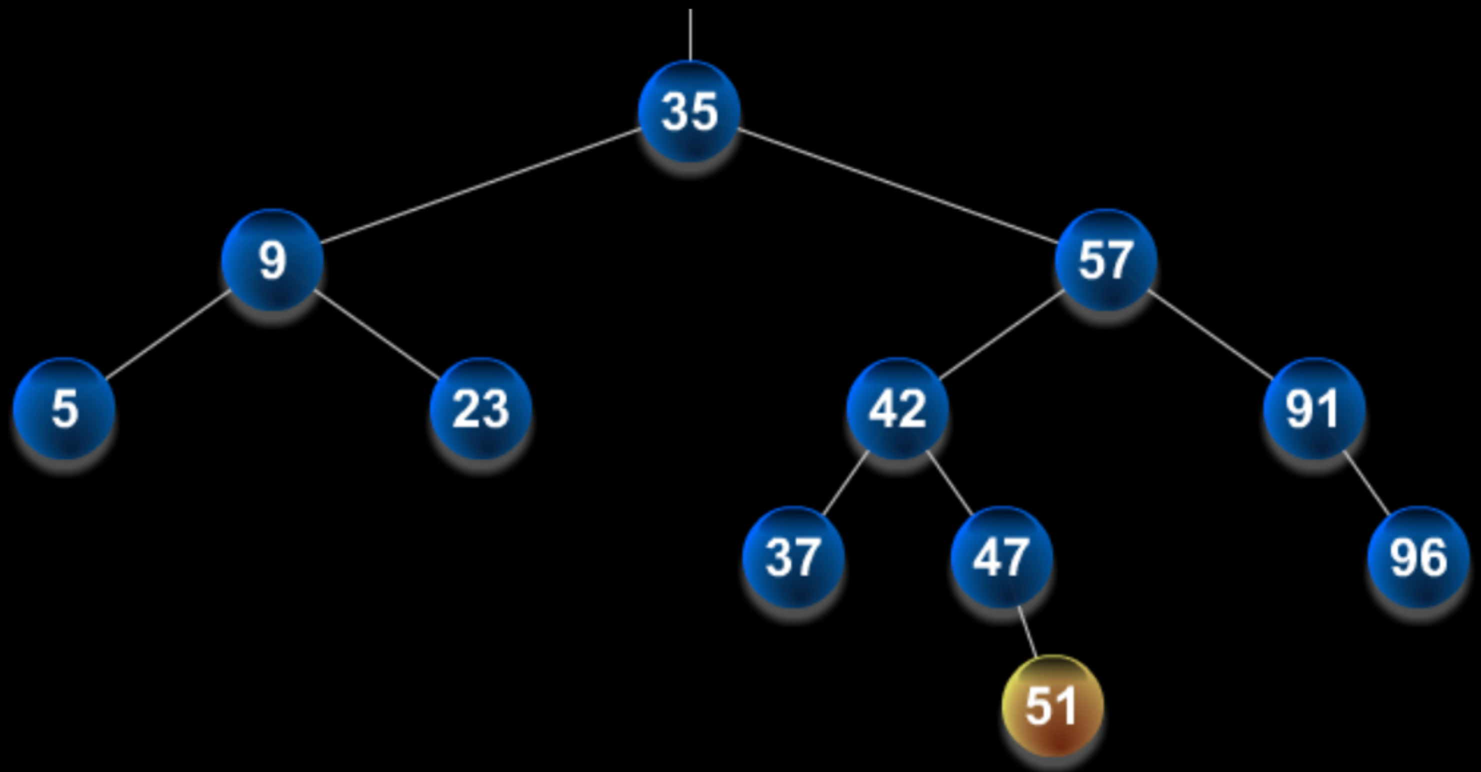
(e)


https://www.cs.tufts.edu/comp/15/schedule/lectures/midterm_info/midterm-review.pdf

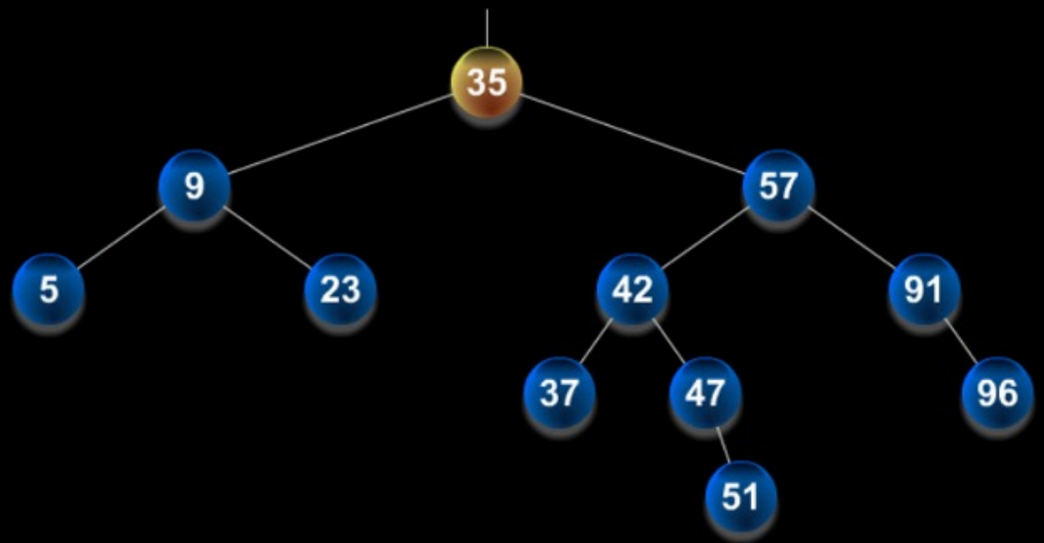
Please Comment your answer on youtube comment box

Question: 2

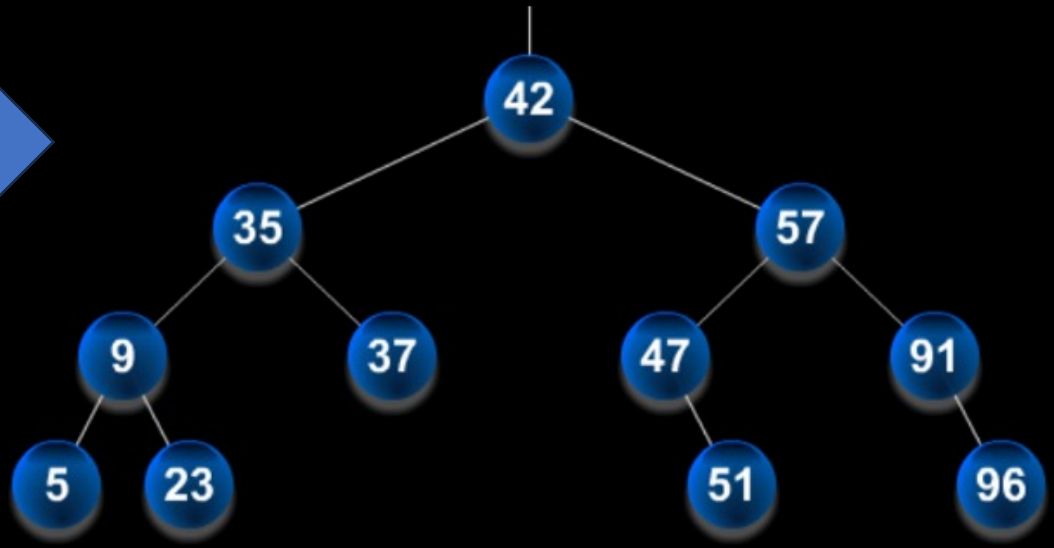
We have just inserted 51, rebalance this tree.



Answer



35 Unbalanced, inserted right-left



Left rotate 35

Question: 3

Which is true of AVL trees?

A. Adding a node may cause the tree to increase in height
B. Adding a node will cause the tree to increase in height
C. Adding a node may cause the tree to decrease in height
D. Adding a node will cause the tree to decrease in height

https://www.cs.ou.edu/~hougen/classes/Fall-2019/DataStructures/materials/OldExams/exam3Fall2015_soln.pdf

Answer A

Question: 4

What is the time complexity for efficiently finding an arbitrary item in an arbitary *AVL* tree of n nodes?

A. $\Theta(n)$
B. $\Theta(\log_2 n)$
C. $\Theta(1)$
D. $\Theta(n^2)$
E. $\Theta(n \log_2 n)$

https://www.cs.ou.edu/~hougen/classes/Fall-2019/DataStructures/materials/OldExams/exam3Fall2015_soln.pdf

Answer B

Question: 5

What is the space complexity for efficiently storing n items in an *AVL* tree?

A. $\Theta(n)$

B. $\Theta(\log_2 n)$

C. $\Theta(1)$

D. $\Theta(n^2)$

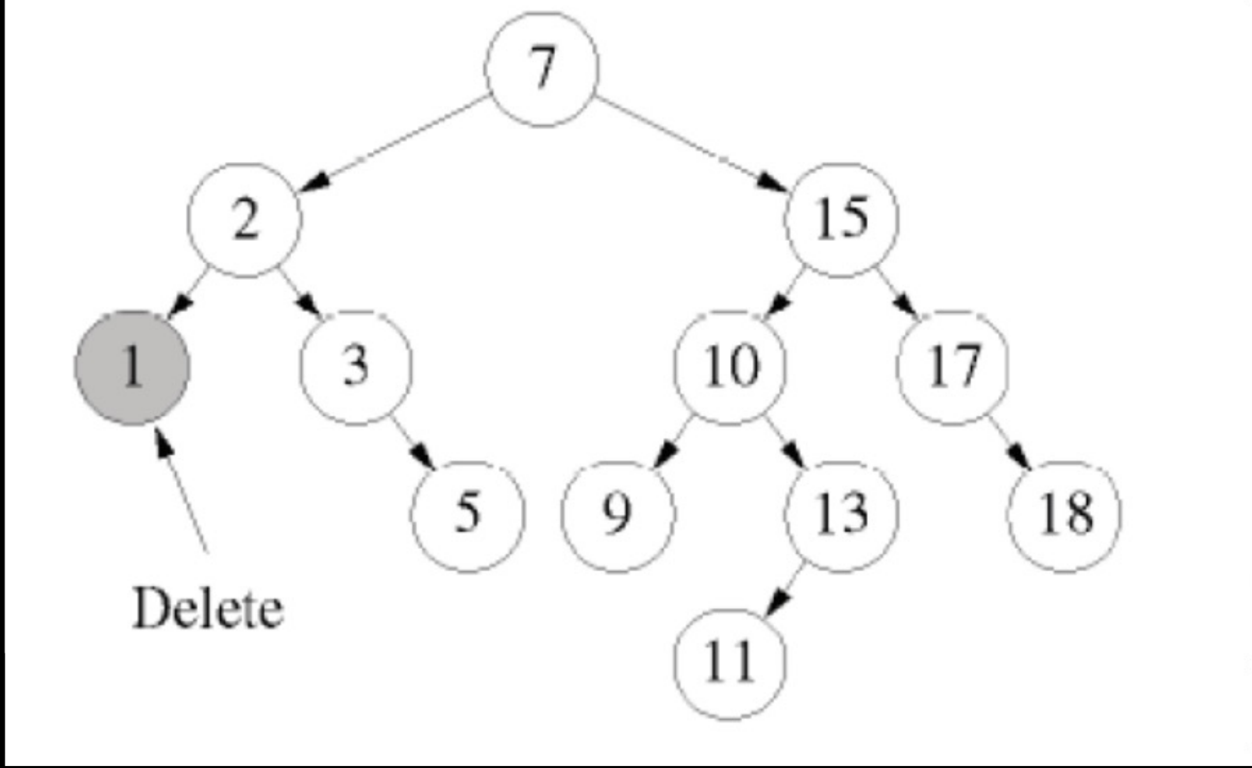
E. $\Theta(n \log_2 n)$

https://www.cs.ou.edu/~hougen/classes/Fall-2019/DataStructures/materials/OldExams/exam3Fall2015_soln.pdf

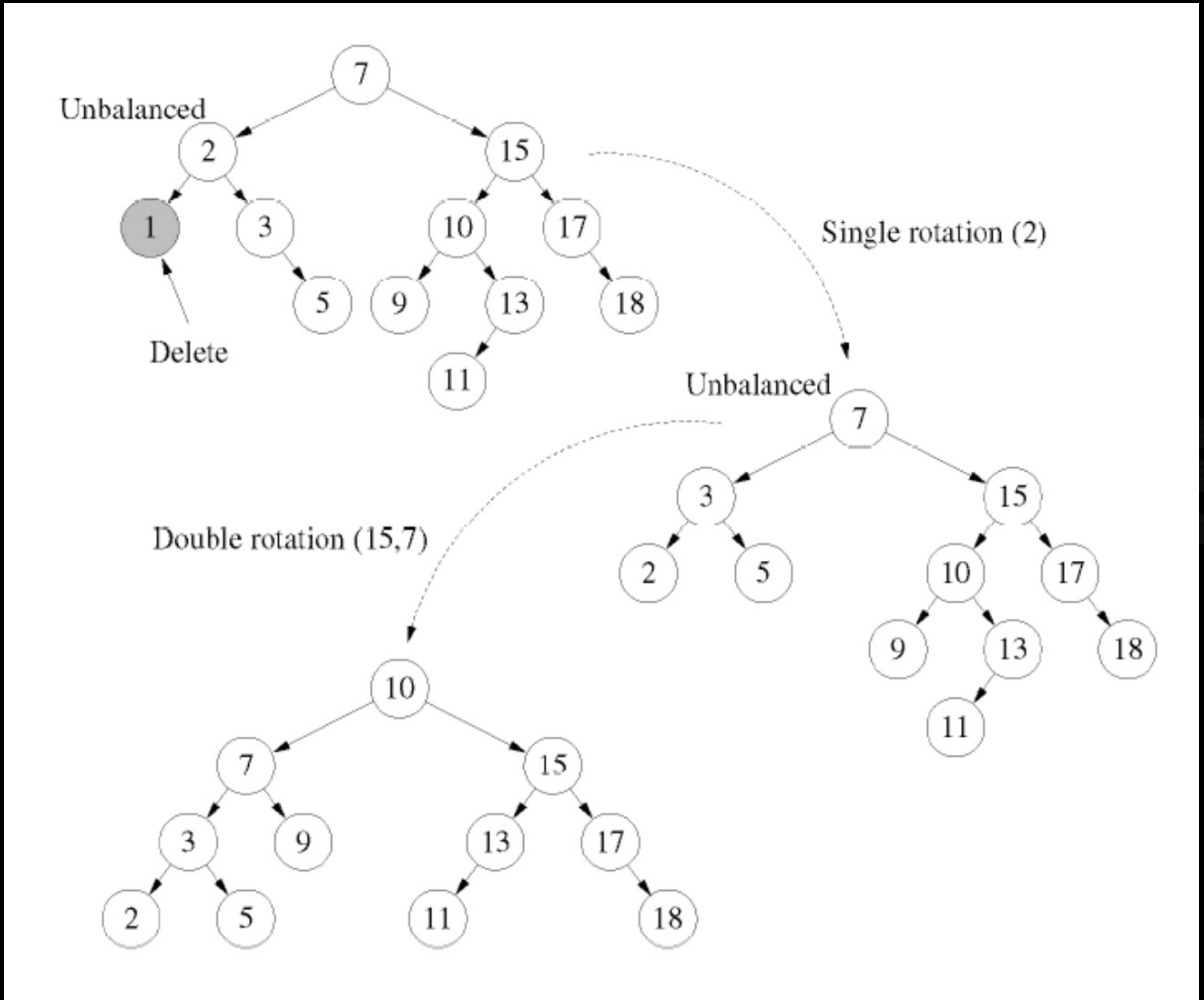
Answer A

Question: 6

Given the AVL tree below, show the AVL tree that would result after deleting the key of value 1.



Answer


Unbalanced
7
2
15
1
3
10
17
5
9
13
18
11
Delete
Single rotation (2)
Unbalanced
7
3
15
2
5
10
17
9
13
18
11
Double rotation (15,7)
10
7
15
3
9
13
17
2
5
11
18

Question: 7

Insertion of a new element into an AVL tree may result in a violation of the balance condition of AVL trees. In such a case a (single or double) rotation restores the balance condition. According to the lectures, a rotation is performed on the first node on the path from the inserted element to the root that is in violation of the AVL balance condition. What would happen if we would instead choose the first node on the path from the root to the inserted element that is in violation of the AVL balance condition?

A. After the rotation, at least one node will be in violation of the balance condition, and thus another rotation will always be necessary.

B. The entire AVL tree will be balanced after the rotation, but more nodes may be changed by the rotation, and thus the rotation may require more time.

C. The entire AVL tree will be balanced after the rotation, and the same number of nodes will be changed.

D. The entire AVL tree will be balanced after the rotation, and the same number or fewer nodes may be changed by the rotation.

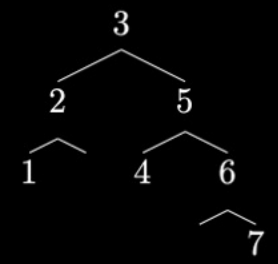
E. The entire AVL tree may or may not be balanced after the rotation.

https://www.comp.nus.edu.sg/~cs1102s/midterms/2009_midterm_2_solution.pdf

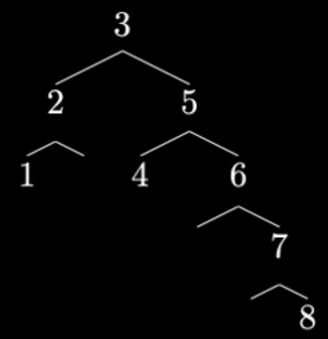
Answer: E

$\boxed{\text{E}}$ If there is only one node that violates the AVL balance condition, both versions pick the same node, and we know that after the rotation the balance condition will hold. Thus if we choose the highest node, the tree may be balanced after the rotation.

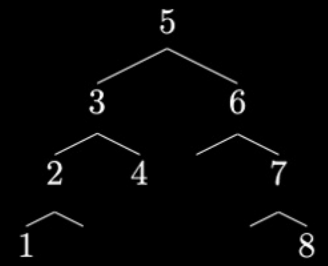
However, the following example shows that the tree may not be balanced after the rotation. Consider insertion of the number 8 into the following tree.



The result of the insertion is the following tree.



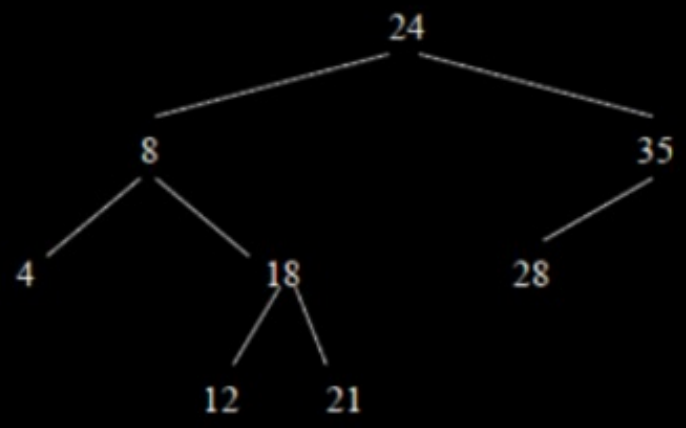
The highest node that violates the balance condition is the root, and rotation at the root leads to the following tree.



This tree still has a node that does not meet the balance condition, namely 6.

Question: 8

Consider the following AVL tree. Show the resulting trees after inserting 10, and then again after deleting 28.

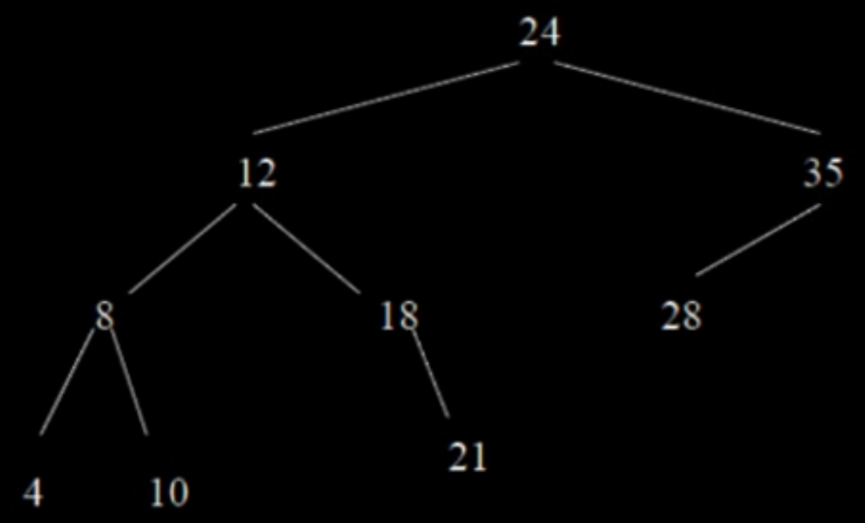


http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-2017-sol.pdf

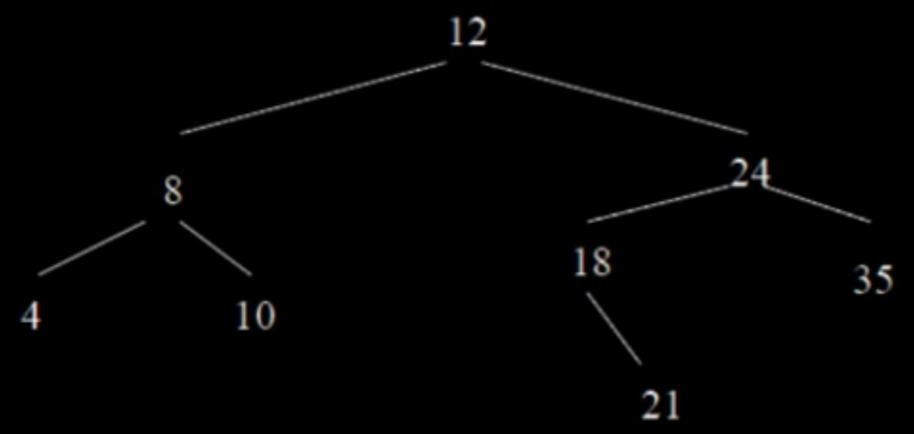
Answer

Ans.

AVL Tree
After inserting 10

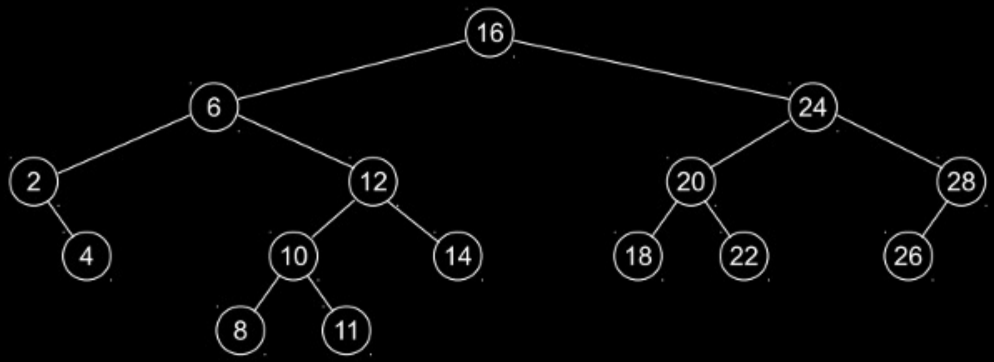


After deleting 28



Question: 9

Given the AVL tree in the following figure:



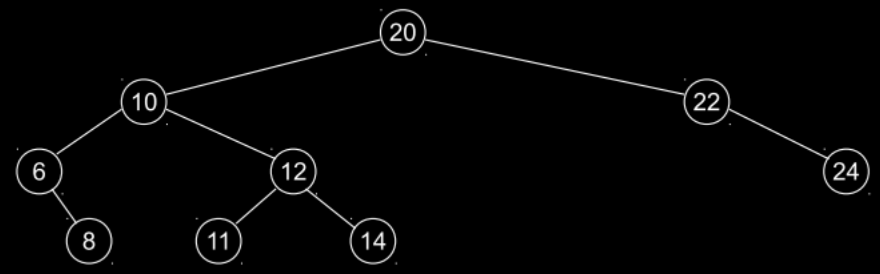
(a) Remove the smallest element twice, then remove the largest element twice, then remove the root node twice.

(b) After completing all removals from part (a), insert the following values in the given order: 17, 1, 15, 36, 41.

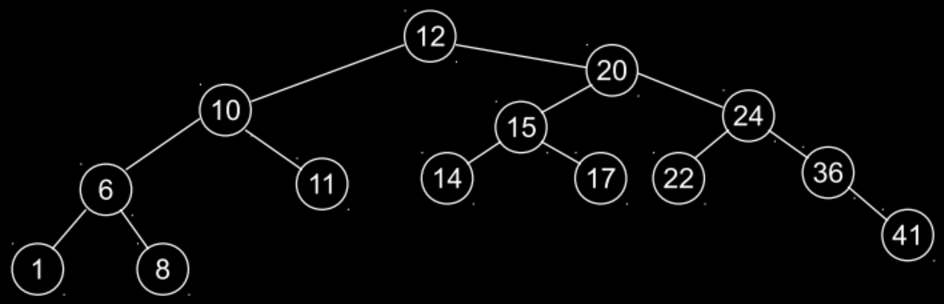
https://ece.uwaterloo.ca/~cmoreno/ece250/ece250-T5_solutions.pdf

Answer

(a)



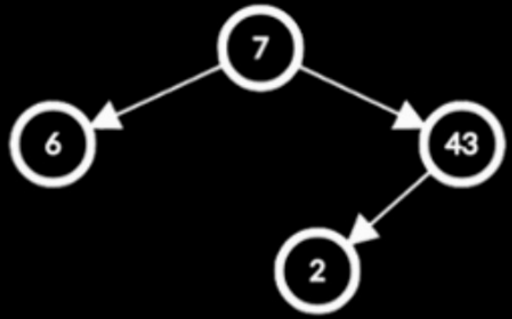
(b)



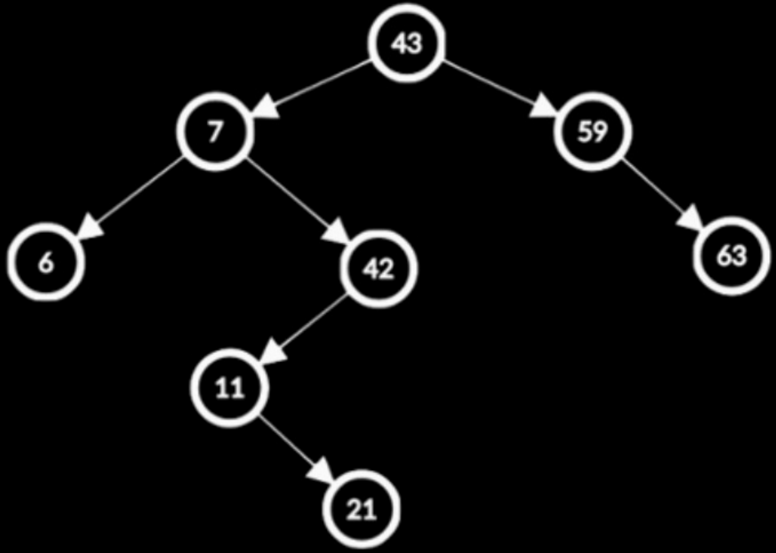
Question: 10

For each of the following trees, state whether the tree is (i) a valid BST and (ii) a valid AVL tree.

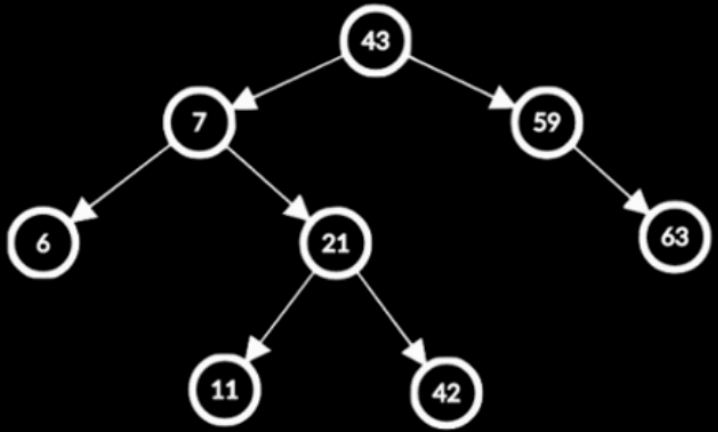
(a)



(b)



(c)



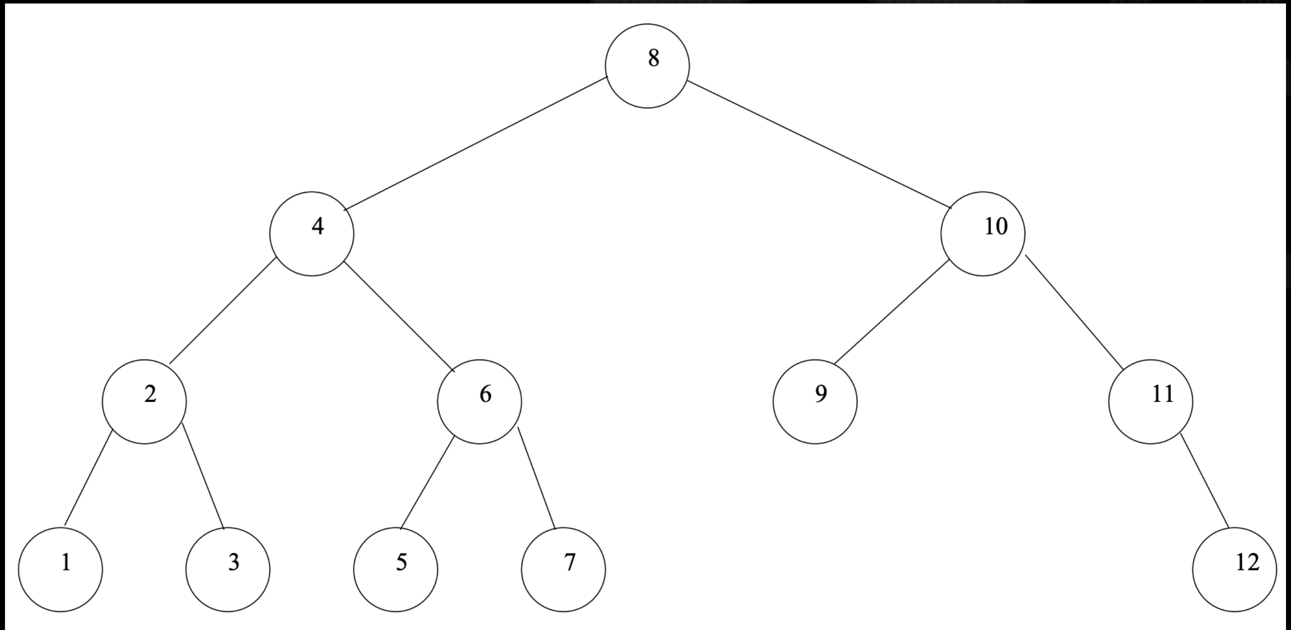
https://courses.cs.washington.edu/courses/cse373/20su/sections/section04-solutions.pdf

Answer :

a) This is not a valid BST!
b) This tree is a valid BST! However, this is not a valid AVL tree.
c) This tree is a valid BST! This tree is also a valid AVL tree!

Question: 11

What will be the minimum number of keys you need to delete from given AVL tree before you cause a double rotation to happen?



https://cs.nyu.edu/~yap/classes/funAlgo/03s/hw/h2/sol2.pdf

Answer : 5

delete(1) delete(3) delete(12) delete(9) delete(11)

For diagrams see last page at

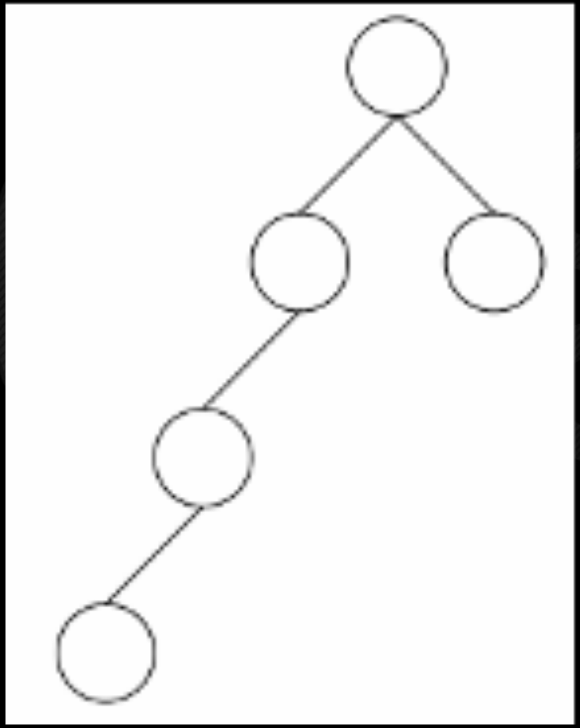
https://cs.nyu.edu/~yap/classes/funAlgo/03s/hw/h2/sol2.pdf

Question: 12

Define a *funny-tree* to be a binary tree such that for each of its nodes x, the number of nodes in each sub-tree of x is at most 2/3 the number of the nodes in the tree rooted at x.

Draw the *tallest* funny tree of 5 nodes.

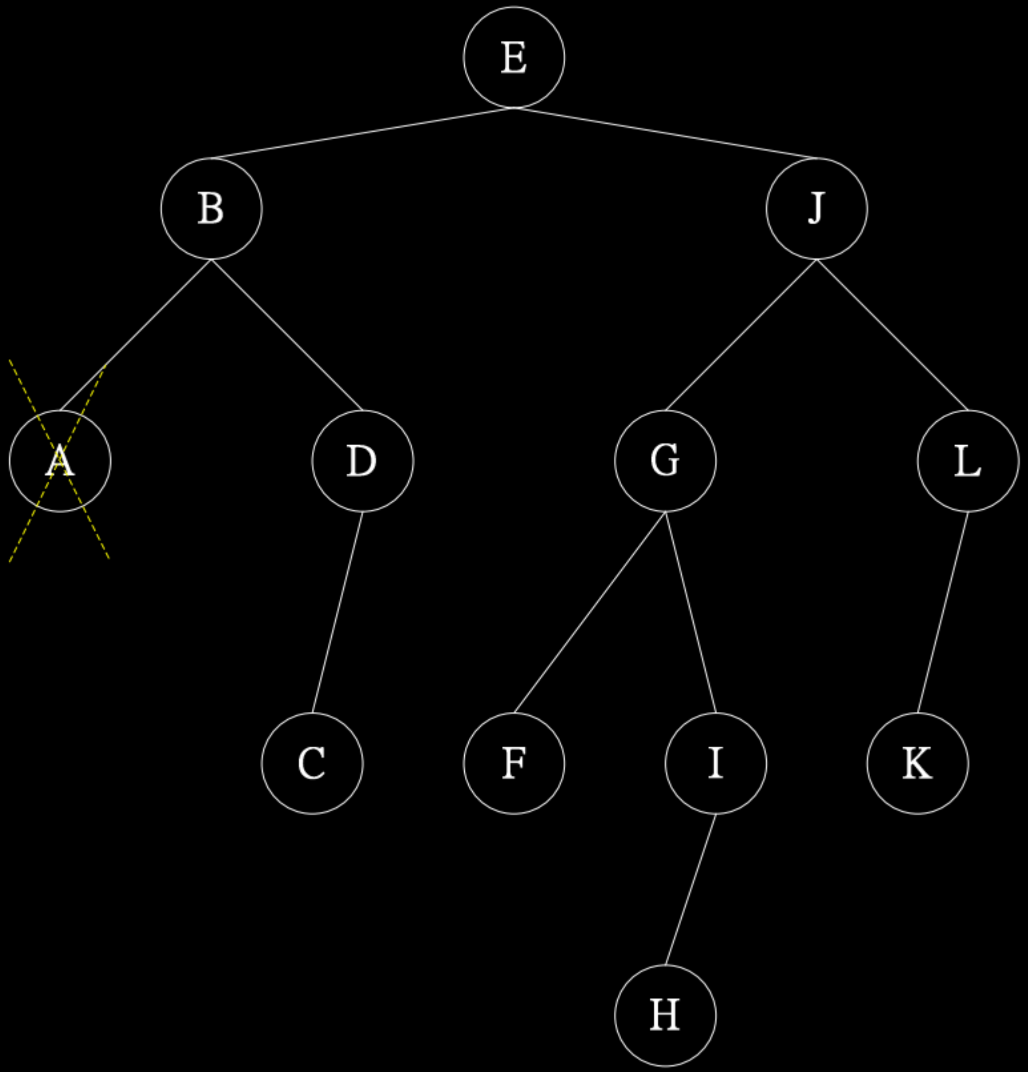
Answer



Question: 13

How many rotations (single or double) are required to rebalance the given AVL tree after deleting node A ?

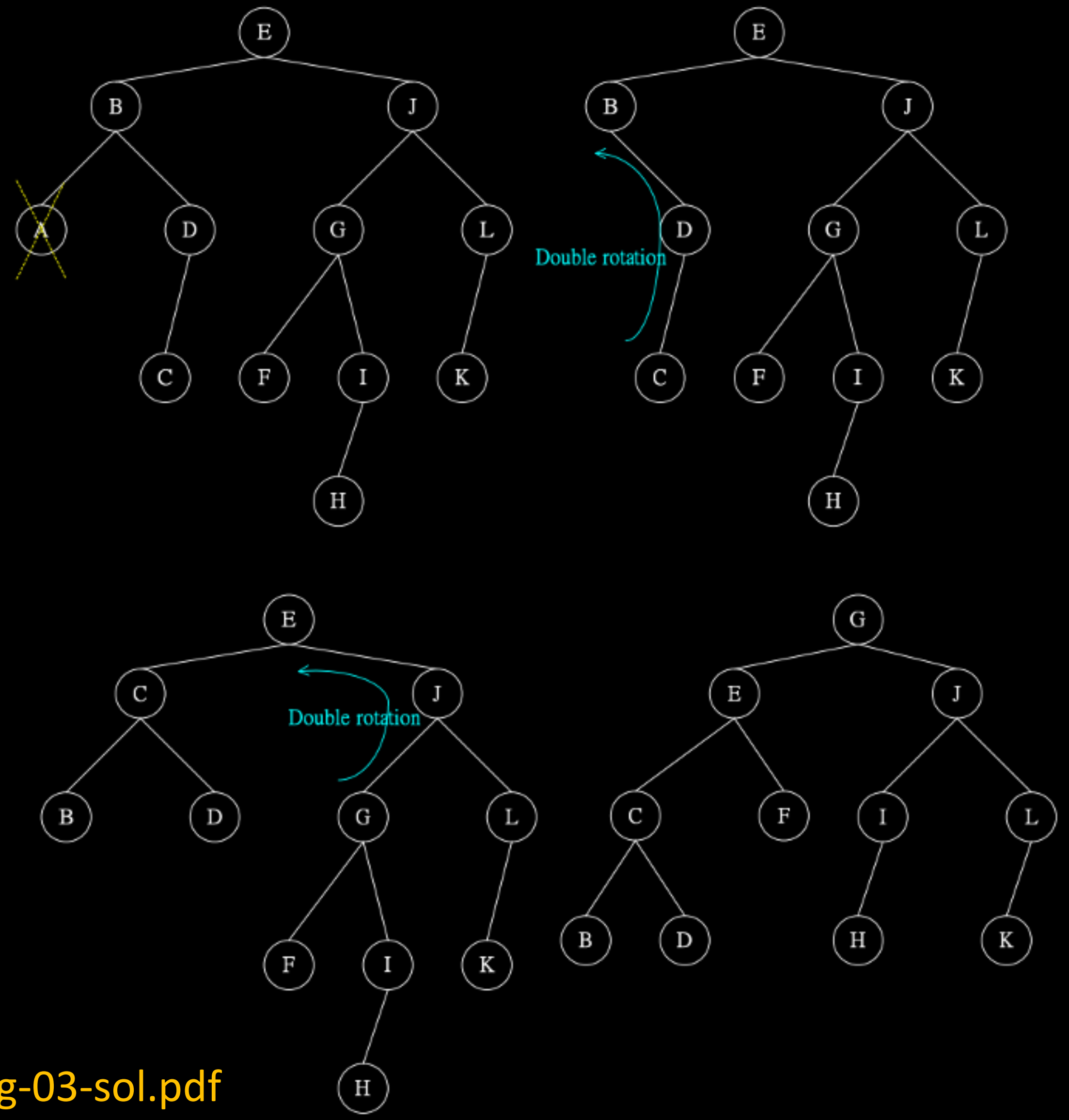
Count each rotation only once, whether it is a single or double rotation.



http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-spring-03.pdf

Answer : 2

8.



http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-spring-03-sol.pdf

Question: 14

Let T be the smallest AVL tree of height h. How many nodes does it have, if the smallest AVL tree of height $h-2$ has m nodes and the smallest AVL tree of height $h-3$ has k nodes?

A. $m + k + 2$
B. $m + 2k$
C. $2m + k$
D. $2m + k + 2$

http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-sol-2010.pdf

Answer : D

Solution:

Let $n(h)$ be the smallest number of nodes in AVL tree of height h.

Then $n(h-1) = 1 + n(h-2) + n(h-3) = 1 + m + k$

and $n(h) = 1 + n(h-1) + n(h-2) = 1 + 1 + m + k + m = 2 + 2m + k$.

Question: 15

Determine the minimum and maximum possible height of an AVL tree with 50 nodes.

http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-fall-03-sol.pdf

Answer :

Minimum height with 50 nodes is 5
Maximum height with 50 nodes is 6

let m(h) and M(h) be the minimum and maximum number of nodes in an AVL tree of height h. If an AVL tree with 50 nodes has height h it must be that $m(h) \leq 50 \leq M(h)$. Therefore, we are looking for the minimum and maximum h that satisfy the above inequalities.

As we have seen m(h) is given by the recurrence

$$m(h) = \begin{cases} 1, & \text{if } h=0 \\ 2, & \text{if } h=1 \\ m(h-1)+m(h-2)+1, & \text{if } h>1 \end{cases}$$

On the other hand, $M(h)=1+2+2^2+\ldots+2^h=2^{h+1}-1$. Therefore,

h	m(h)	M(h)
0	1	1
1	2	3
2	4	7
3	7	15
4	12	31
5	20	63
6	33	127
7	54	255

So, the minimum and maximum h that satisfy $m(h) \leq 50 \leq M(h)$ are 5 and 6.

Question: 16

Answer the following 4 questions in theta (θ) notation

The ***minimum*** number of *rotations* needed to perform an *insertion* on an AVL tree of n nodes.

The ***minimum*** number of *rotations* needed to perform a *deletion* on an AVL tree of n nodes.

The ***maximum*** number of *rotations* needed to perform an *insertion* on an AVL tree of n nodes.

The ***maximum*** number of *rotations* needed to perform a *deletion* on an AVL tree of n nodes.

http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-fall-03.pdf

Please Comment your answer on youtube comment box

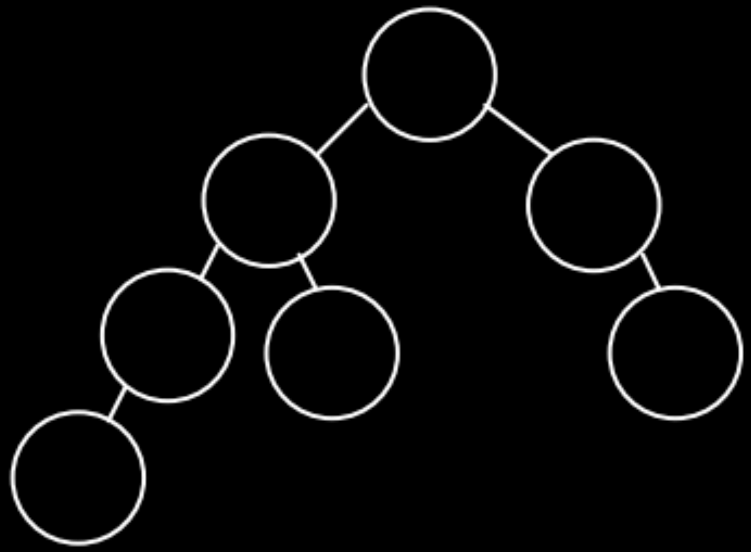
Question: 17

Draw an AVL tree of height 3 with a minimum number of nodes.

http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-Sping07.pdf

Answer :

7 nodes in total:



Question: 18

Suppose we define a variation on AVL trees in which the heights of the left and right subtrees of a node could differ by at most 2 (rather than at most 1).

What recurrence relation would we get for the minimum number n_k of keys in a tree of height k?

A. $n_k = n_{k-1} + n_{k-2} + 1$

B. $n_k = n_{k-1} + n_{k-3} + 1$

C. $n_k = n_{k-1} + n_{k-2} + 2$

D. $n_k = n_{k-1} + n_{k-3} + 3$

http://ccf.ee.ntu.edu.tw/~yen/courses/ds17/midterm-Sping07.pdf

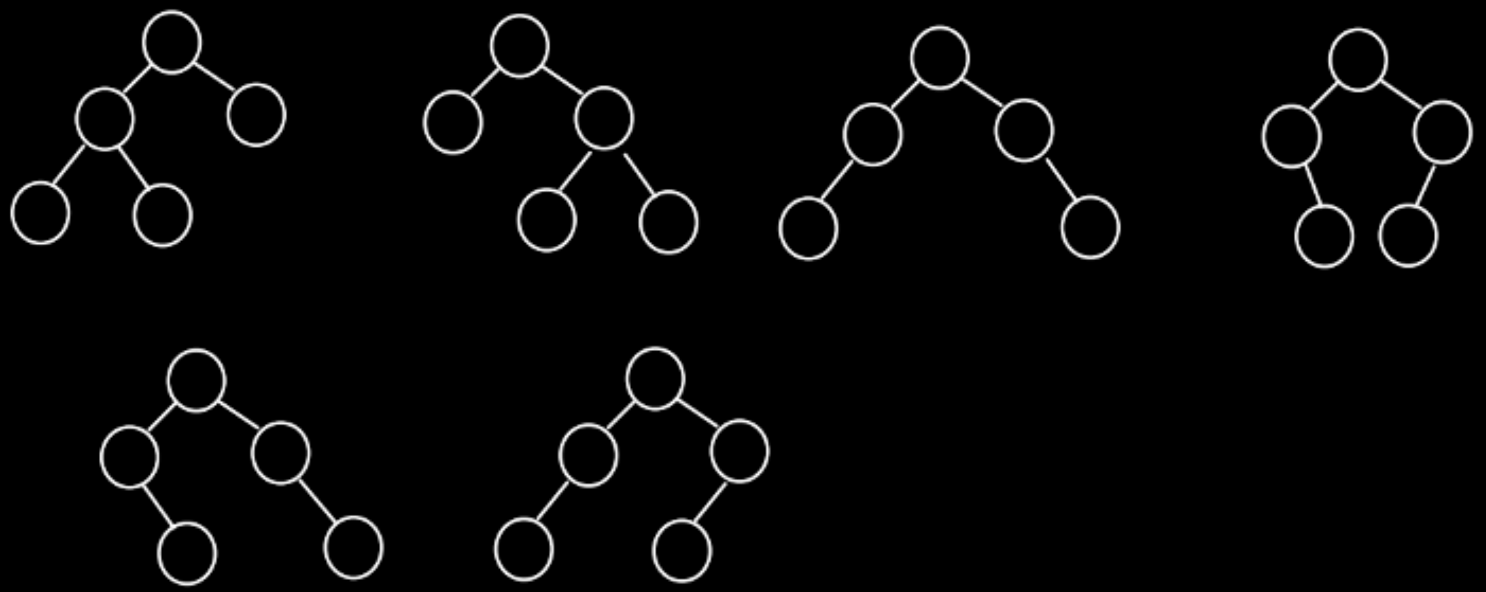
Answer : B

Question: 19

Draw the structures of all AVL-trees with 5 nodes.

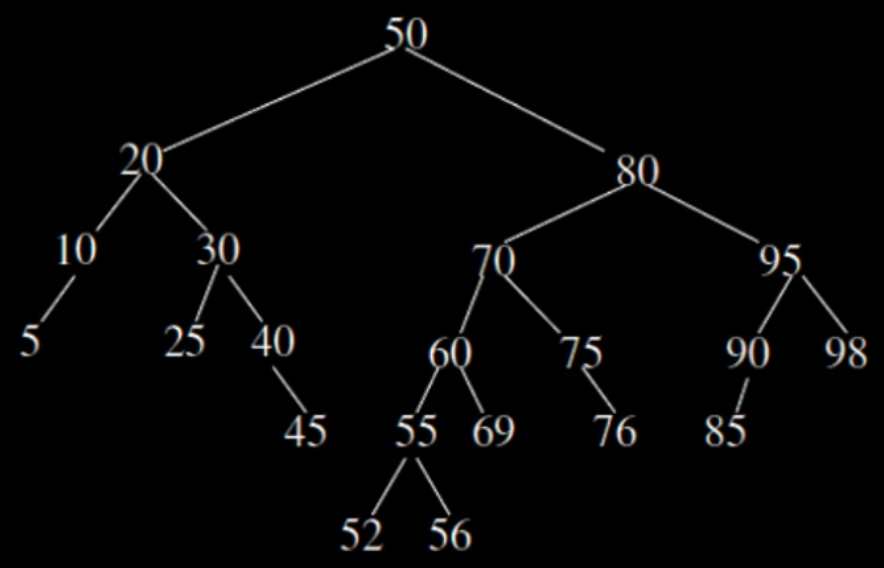
http://ccf.ee.ntu.edu.tw/~yen/courses/ds20F/midterm-2020-sol.pdf

Answer :



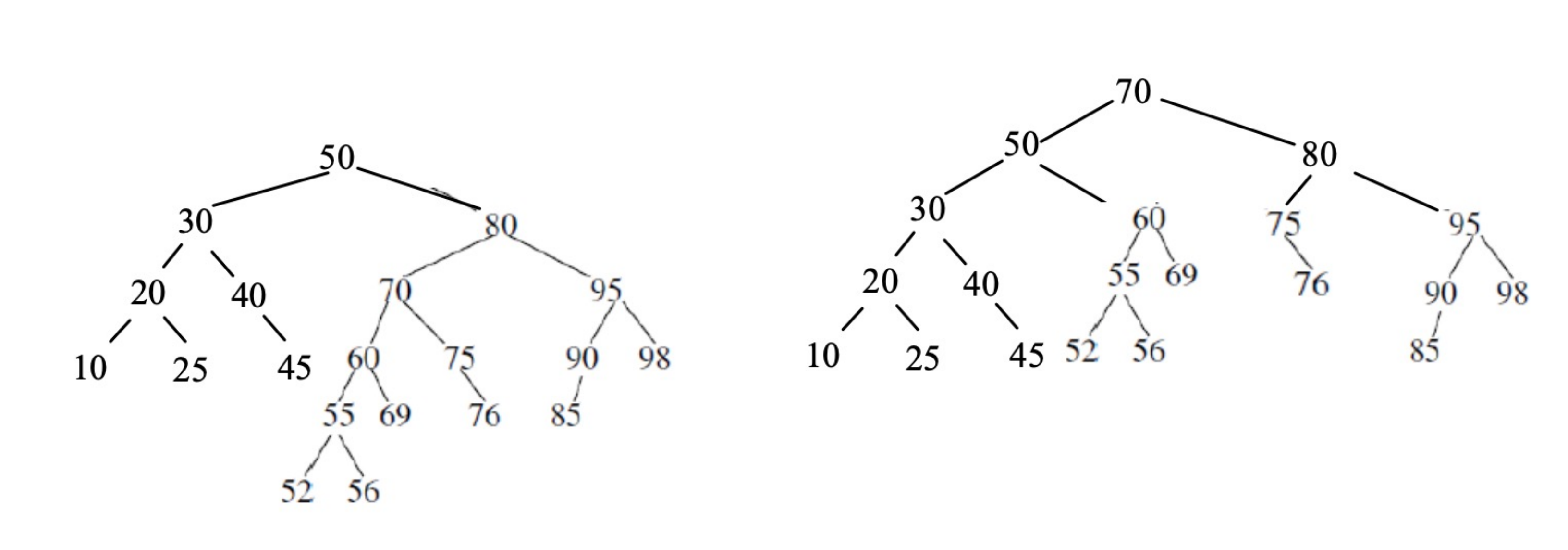
Question: 20

Delete element 5 in the following AVL tree and rebalance the tree. Show (i.e., draw) each of the intermediate trees after a single or a double-rotation is carried out.



http://ccf.ee.ntu.edu.tw/~yen/courses/ds20F/midterm-2020-sol.pdf

Answer :


50
30
80
20
40
70
95
10
25
45
60
75
90
98
55
69
76
85
52
56
70
50
80
30
60
75
95
20
40
55
69
76
90
98
10
25
45
52
56
85

Question: 21

An AVL tree of height h has a minimum number of nodes x and an AVL tree of height $(h - 2)$ has a minimum number of nodes y.

What is the minimum number of nodes for an AVL tree of height $(h + 1)$?

Assume height h is an integer greater than 2.

a) x+y+1

c) 2x-y

b) 2x+y+1

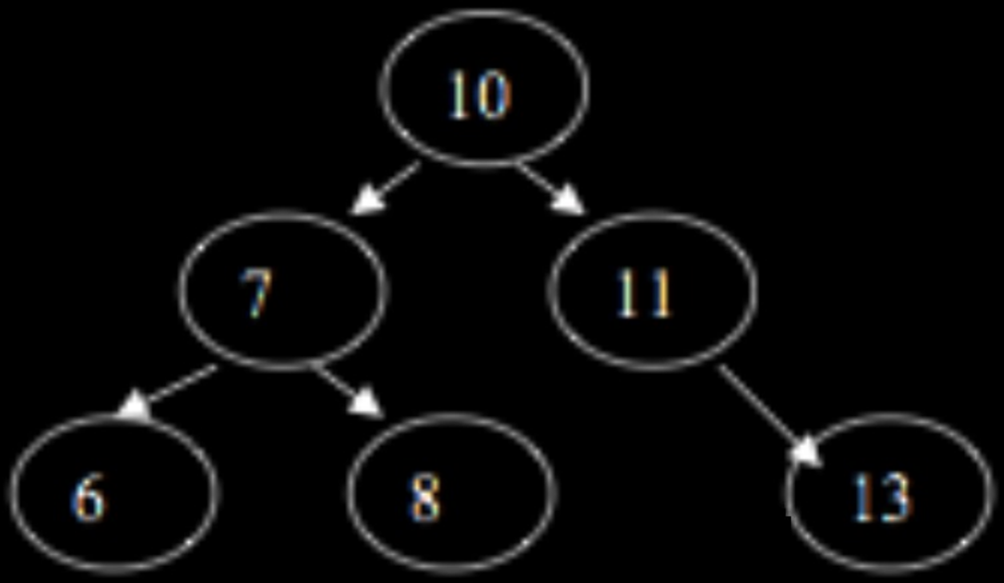
d) x-y+2

https://courses.cs.washington.edu/courses/cse332/23sp/exams/oldExams/cse332-midterm-23wi-soln.pdf

Answer : B

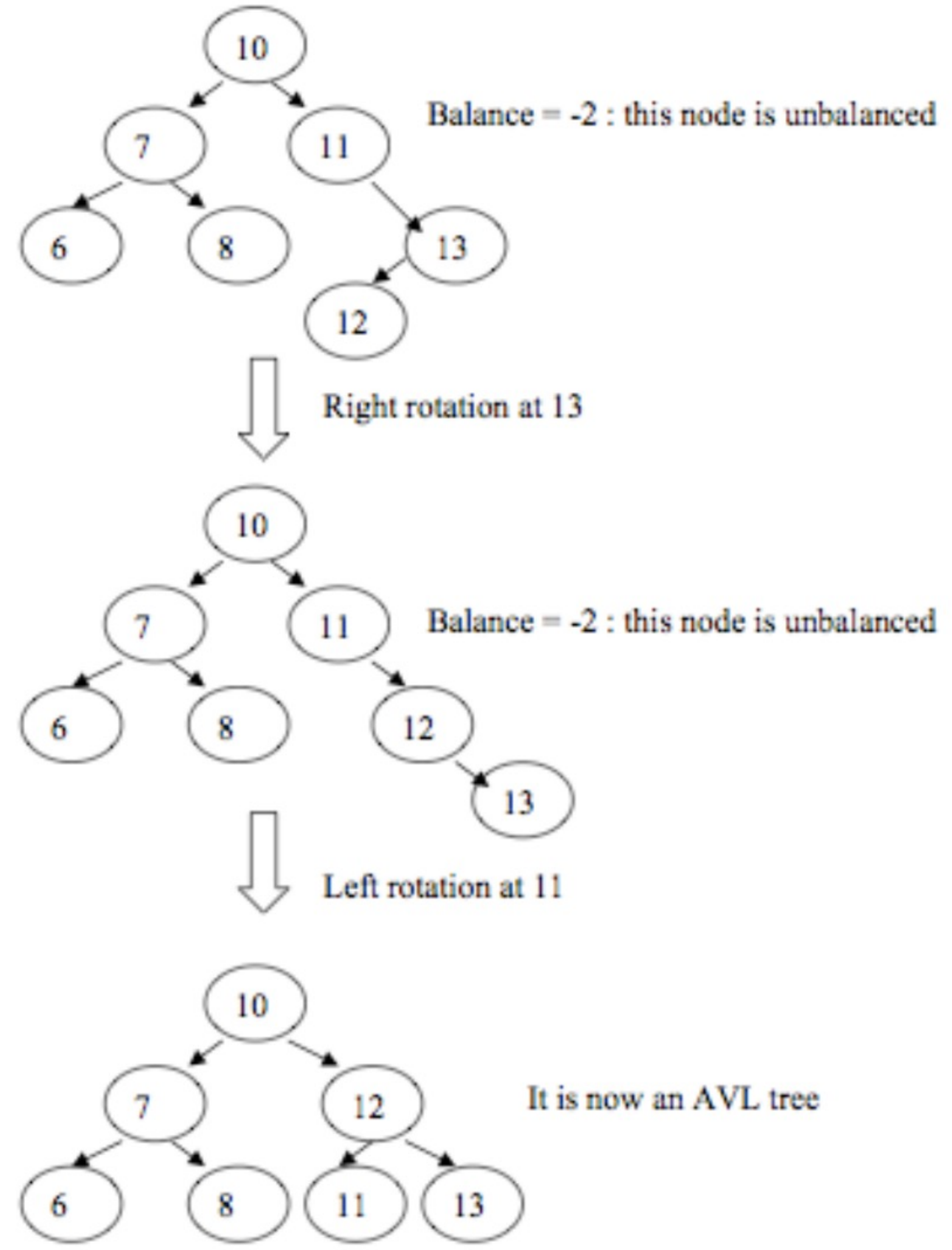
Question: 22

Consider the following AVL tree containing distinct keys.
Which integer number, distinct from the ones already in the AVL tree, should be inserted to maintain the same height of the tree?



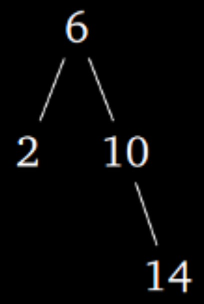
https://courses.engr.illinois.edu/cs225/fa2017/exams/practice/fa07mt2soln.pdf

Answer : 12


10
7
11
6
8
13
12
Balance = -2 : this node is unbalanced
Right rotation at 13
10
7
11
6
8
12
13
Balance = -2 : this node is unbalanced
Left rotation at 11
10
7
12
6
8
11
13
It is now an AVL tree

Question: 23

Consider this AVL tree:



Give an example of a value you could insert to cause:

(a) A single rotation

(b) A double rotation

(c) No rotation

https://courses.cs.washington.edu/courses/cse373/20su/sections/section04-solutions.pdf

Answer :

(a) A single rotation

Solution:

Any value greater than 14 will cause a single rotation around 10 (since 10 will become unbalanced, but we'll be in the line case).

(b) A double rotation

Solution:

Any value between 10 and 14 will cause a double rotation around 10 (since 10 will be unbalanced, and we'll be in the kink case).

(c) No rotation

Solution:

Any value less than 10 will cause no rotation (since we can't cause any node to become unbalanced with those values).

Question: 24

We want to insert the following sequence of keys into an initially empty search tree:

$$1, -1, \frac{1}{2}, \frac{-1}{2}, \frac{1}{3}, \frac{-1}{3}, \ldots, \frac{1}{n}, \frac{-1}{n}.$$

Let $n = 5$ (so you are inserting 10 keys). Draw the AVL tree at the *end* of each insertion. So we want to show us 10 trees for this question. You get one point per tree, and if any tree is wrong, then all subsequent trees are considered wrong. Do NOT show us your intermediate results, although you probably want to do this on the scratch pages. Please be careful, as it is easy to make mistakes.

https://cs.nyu.edu/~yap/classes/funAlgo/03s/hw/midterm/mid.pdf

Answer :

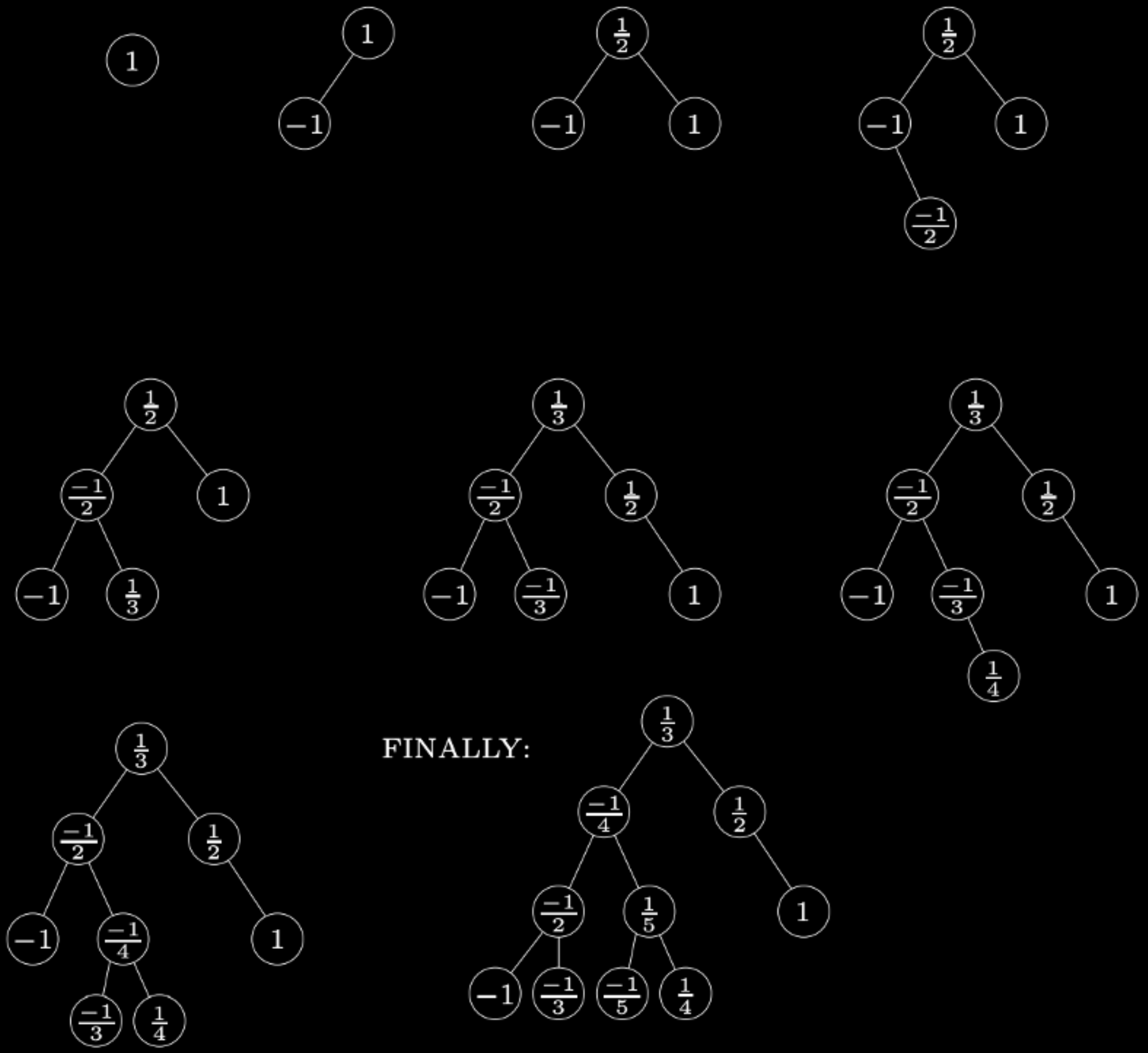
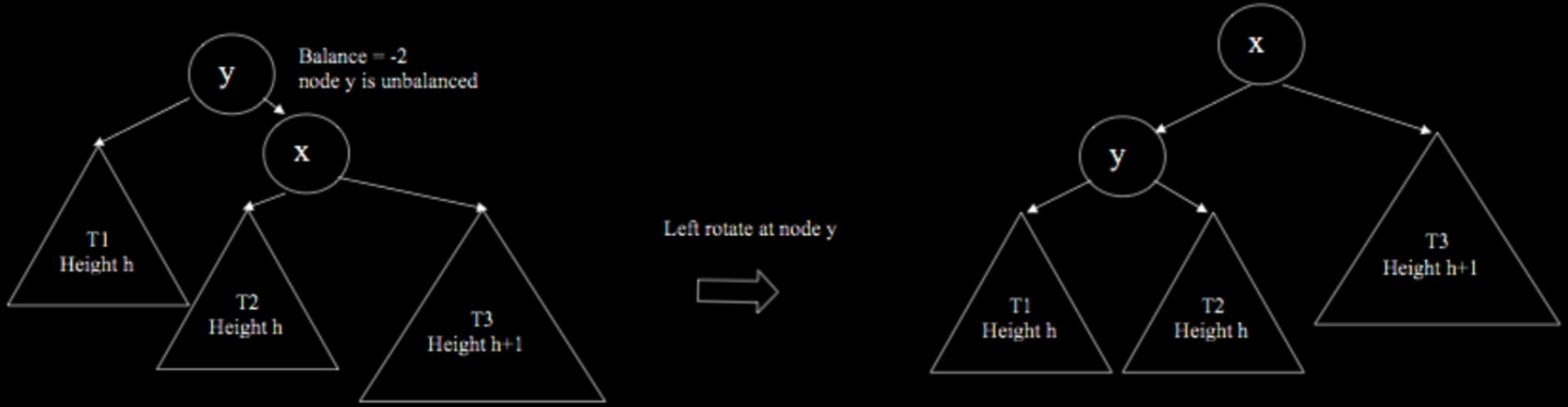


Figure 2: Inserting $1, -1, 1/2, -1/2, , \ldots, 1/5, -1/5$ into an AVL tree.

Question: 25

We are faced with a situation where we need to perform a rotation to address an imbalance at node y. To accomplish this, we have created a function called `rotateLeft`, which accepts a pointer to the y node.

(function is on the next page)



https://courses.engr.illinois.edu/cs225/fa2017/exams/practice/fa07mt2soln.pdf

```
struct AVLTreeNode rotateLeft(struct AVLTreeNode *node) {

    struct AVLTreeNode *y = node;

     //Fill the missing code here

     y->height = max(y->left->height, y->right->height)+1;
     node->height = max(node->left->height, node->right->height)+1;


    return node;
}
```

Can you suggest the correct options to fill in the missing lines ?

A.
```
node->left = y;
node = node->right;
y->right = node->left;
```

B.
```
node = node->right;
y->right = node->left;
node->left = y;
```

C.
```
node = node->left;
y->right = node->right;
node->left = y;
```

D.
```
y->right = node->left;
node = node->right;
node->left = y;
```

Answer : B

Question: 26

We have an AVL tree that contains the integers $1, 2, 3, 4, 5, 6, 7$.

How many valid AVL trees can be formed with those numbers?

There's a simpler way to count than enumerating each valid AVL tree one by one. It will probably take too much time to enumerate all of them manually.

Hint: think about the possible values the root can take.

https://courses.cs.washington.edu/courses/cse332/23sp/exams/oldExams/cse332-midterm-23wi-soln.pdf

Answer : 17

Solution:

17. The valid values at the root are $3, 4, 5$. When root is 3 or 5, the number of valid combinations are equal (symmetric) - in other words, one child needs to have 2 nodes, and the other child needs to have 4 nodes. An AVL tree with 2 nodes can only have a height of 1 (can draw it out to validate), and an AVL tree with 4 nodes can only have a height of 2 (can draw it out too to validate). An AVL tree with 2 nodes has 2 combinations, and an AVL tree with height 4 has 4 combinations. So $2 * 4 * 2 = 16$. Note the 2 at the end accounts for the symmetric case when root is 3 and 5. When root is 4, left and right child has to have exactly 3 nodes, and there's only 1 combination to lay out a valid AVL tree with height 1. So overall, we have $16 + 1 = 17$ valid variations.

Question: 27

If we have an AVL Tree with 11 nodes, what is the maximum height of the AVL Tree?

https://courses.cs.washington.edu/courses/cse332/23sp/exams/oldExams/cse332-midterm-22su-soln.pdf

Answer : 3

Solution:

The maximum height is 3.

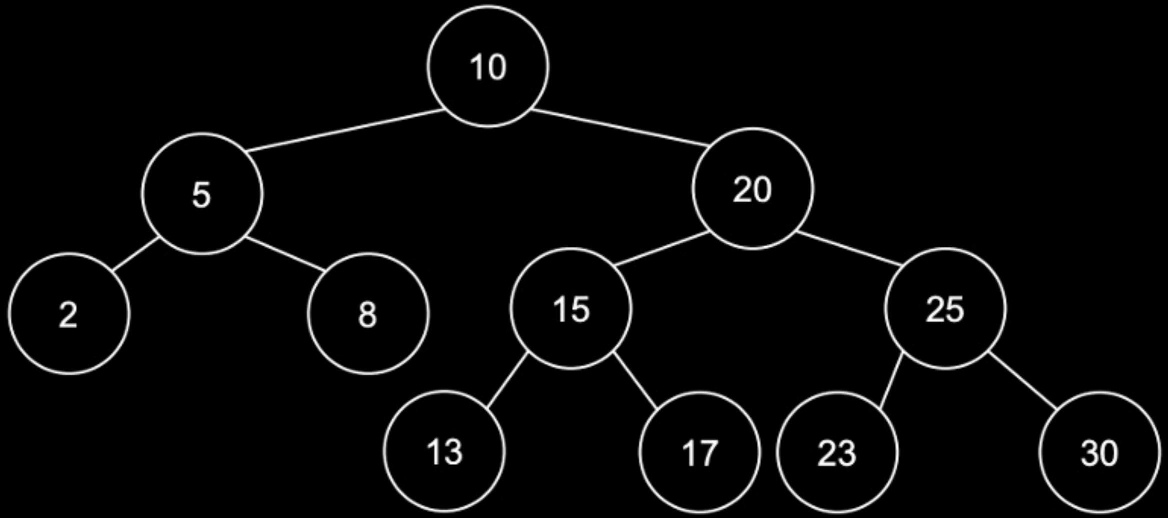
We can calculate this by using the formula $S(h) = S(h-1) + S(h-2)$.

$$S(0) = 1 S(1) \qquad = 2 S(2) = 2 + 1 + 1 = 4 S(3) \qquad = 4 + 2 + 1 = 7 S(4) = 7 + 4 + 1 = 12$$

So, we know that we need to have at least 12 nodes for a height of 4 and thus 11 can only have a maximum height of 3.

Question: 28

For the following AVL tree:



What values could you insert to cause a right-right imbalance, and at which node does the imbalance occur?

https://courses.cs.washington.edu/courses/cse332/15su/sections/04/section.pdf

Answer :

Inserting any value greater than 20 causes a right-right imbalance at node 10, the root, because the root's left child now has height 1 and the right child has height 3.

https://courses.cs.washington.edu/courses/cse332/15su/sections/04/section-solutions.pdf

Question: 29 MSQ

If you insert 7 elements into an AVL tree, what are the possible heights of the tree?

A. 2
B. 3
C. 4
D. 5

https://courses.cs.washington.edu/courses/cse373/18au/files/sections/section05-solutions.pdf

Answer : A,B

Solution:

An AVL tree with 7 elements could be height 2 or 3. It cannot be height 4: if the height is 4, then for the root to be balanced, one subtree must have height 3 and the other at least 2. A height 2 AVL tree needs at least 4 elements (or the root of the subtree won't be balanced), so in our 7 element tree, we have 4 elements for one subtree, plus the root, which leaves only 2 elements for the other subtree. But that subtree was supposed to be height 3, so there just aren't enough elements to fill it out.

Notice that the AVL tree doesn't guarantee you the minimum possible height (there are AVL trees with 7 elements of height 3), but it does avoid the worst case.

Question: 30

What does the “AVL” stand for in AVL tree?

Answer :

AVL tree is named after its two Soviet inventors, Georgy **A**delson-**V**elsky and Evgenii **L**andis