# W10 -AVL Trees

**Due** 17 Mar at 23:59 **Points** 20 **Questions** 8
**Available** 13 Mar at 5:00 - 17 Mar at 23:59 **Time limit** None
**Allowed attempts** Unlimited

## Instructions

## Content and Background

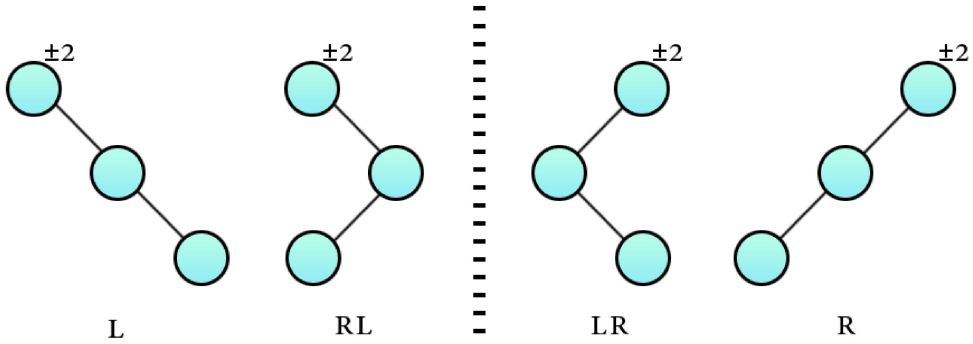
This quiz relates to the content covered in the course up till now. It may also draw upon supporting knowledge and skills expected from a CS sophomore. Please make sure that you are up to date on the coursework before attempting the quiz.

## Difficulty

This quiz is equivalent to an in-class exercise. Have pen and paper ready and be prepared to work on challenging problems.

## Discussion

Please use discussion forums to discuss any of the questions. Do not reveal your answers.




The four cases of imbalance in an AVL Tree, for reference.

Take the quiz again

## Attempt history

| | Attempt | Time | Score |
|---|---|---|---|
| **KEPT** | Attempt 4 | 1 minute | 20 out of 20 |
| **LATEST** | Attempt 4 | 1 minute | 20 out of 20 |

| Attempt | Time | Score |
|---|---|---|
| Attempt 3 | less than 1 minute | 17 out of 20 |
| Attempt 2 | 2 minutes | 15 out of 20 |
| Attempt 1 | 7 minutes | 15 out of 20 |

Correct answers are hidden.

Score for this attempt: **20** out of 20

Submitted 17 Mar at 0:33

This attempt took 1 minute.

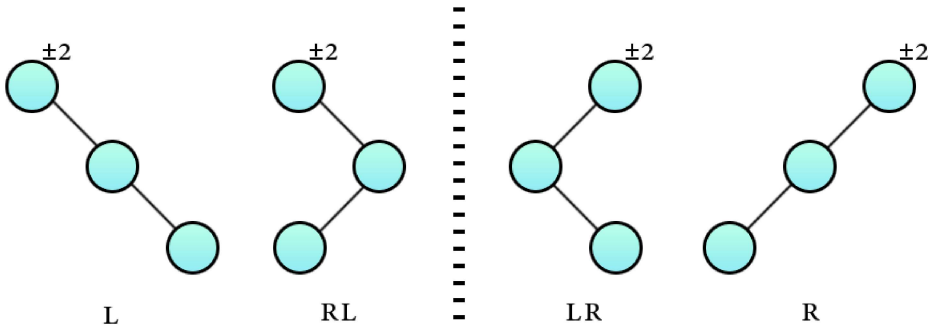
## Question 1

2 / 2 pts

A Treap is expected to build faster than an AVL Tree when... (choose all that apply)

- [ ] The keys to be inserted have high variance
- [x] The keys to be inserted are in random order
- [ ] The keys to be inserted are in sorted order

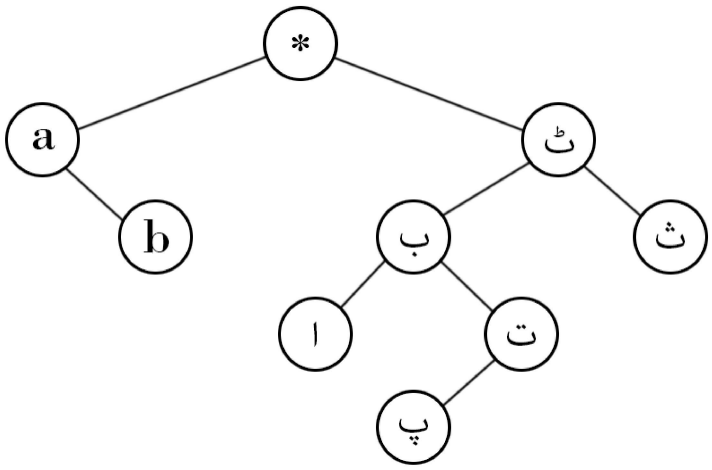
The following questions refer to the four cases of imbalance in an AVL Tree that are illustrated below. The label of each case indicates the type and quantity of rotation operations required to restore balance. The imbalanced node is referred to as *u*.




## Question 2

3 / 3 pts

The following can be converted to an AVL Tree by performing one of the four rotations R, LR, L, RL. Which one is it?




- LR
- R
- RL
- L

## Question 3

2 / 2 pts

In the unbalanced tree above, the first rotation when restoring balance will be at...

- ب
- ت
- پ
- b

## Question 4

3 / 3 pts

The following function is called on a node, *u*, in an AVL tree that is known to be imbalanced. Fill in the blanks in the pseudo-code. (A blank that appears all by itself indicates an entire line of code is to be provided)

```
balance(u):

if is_taller(u.right, u.left) then
   if is_taller(u.right.right, u.right.left) then
      rotate_[left]

(u)#case L
   else
      rotate_right([u.right]

)#case RL
      rotate_left(u)
else
   if is_taller([u.left.right]

, u.left.left) then
      rotate_left(u.left)#case LR
      [rotate_right(u)]


   else
```

```
    rotate_right(u)#case R
```

**Answer 1:**

left

**Answer 2:**

u.right

**Answer 3:**

u.left.right

**Answer 4:**

rotate_right(u)

## Question 5

3 / 3 pts

Match the time complexities with the correct AVL Tree functions. Assume a tree of size **n**.

| | |
|---|---|
| **find** | O(log n) |
| **add** | O(log n) |
| **remove** | O(log n) |

## Question 6

2 / 2 pts

The complexity of a single rotation in an AVL tree is:

- (●) O(1)
- ( ) O(n)
- ( ) O(logn)

## Question 7

3 / 3 pts

The rotation operation in an AVL tree exhibits the following properties:

- [x] The node being rotated ends up with a new parent.
- [ ] After rotation, the tree may not be a binary search tree.
- [ ] It retains the height at the position where the rotation occurred.
- [x] It shifts the height from one side to other, resulting in balancing the tree.

## Question 8

2 / 2 pts

If an insertion in AVL tree results in unbalancing the tree, you would have to perform rotation in:

- (●) the branch in which the node has been inserted.
- ( ) the shallowest branch of the tree.
- ( ) the branch that contains the successor of the newly inserted node.

- the longest branch of the tree.

Quiz score: **20** out of 20