Splay Tree

Course: Algorithms and Data Structures
Teacher: Luis Talavera

Students:
- Juan Gálvez
- Luis Jáuregui
- Nelson Soberón

Description / Properties

- It’s a type of BST.
- Most of the operations take O(log n) amortized time complexity, but in worst case scenarios, they take roundly O(n).
- It follows the same concept as Self Lists with Move-To-Front method that we saw in earlier classes.
- It’s faster than AVL Trees, so that’s the reason this structure is widely used in the industry.
- It’s all about rotations!

Pros & Cons

Pros:

1. Average-case performance is as efficient as other trees.
2. Doesn't need additional memory nor to store tracking data.
3. It's simple to implement.
4. Doesn't require complex operations to balance.

Cons:

1. Since it doesn't care about balance, it may end up being linear, leading to a worst case O(n) Linked-List.
2. The worst case in the above point would be accessing all elements in non-decreasing order.
3. It's complicated to use this structure in a multi-threaded environment, because even when only reading you need to "splay".

Methods

SEARCH/FIND

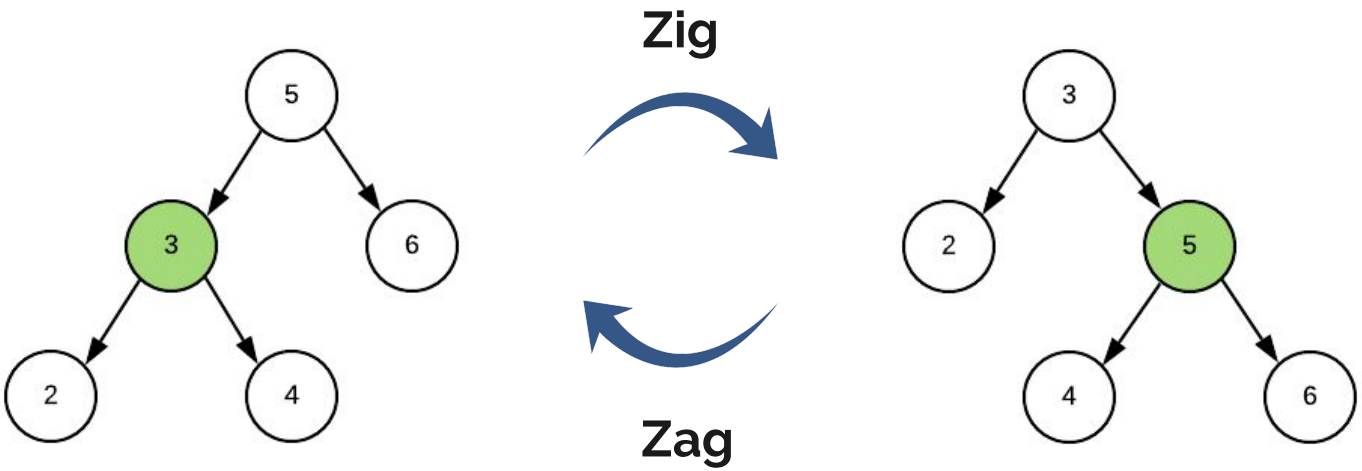
- Zig rotation
- Zig-Zig rotation
- Zig-Zag rotation

INSERT

DELETE

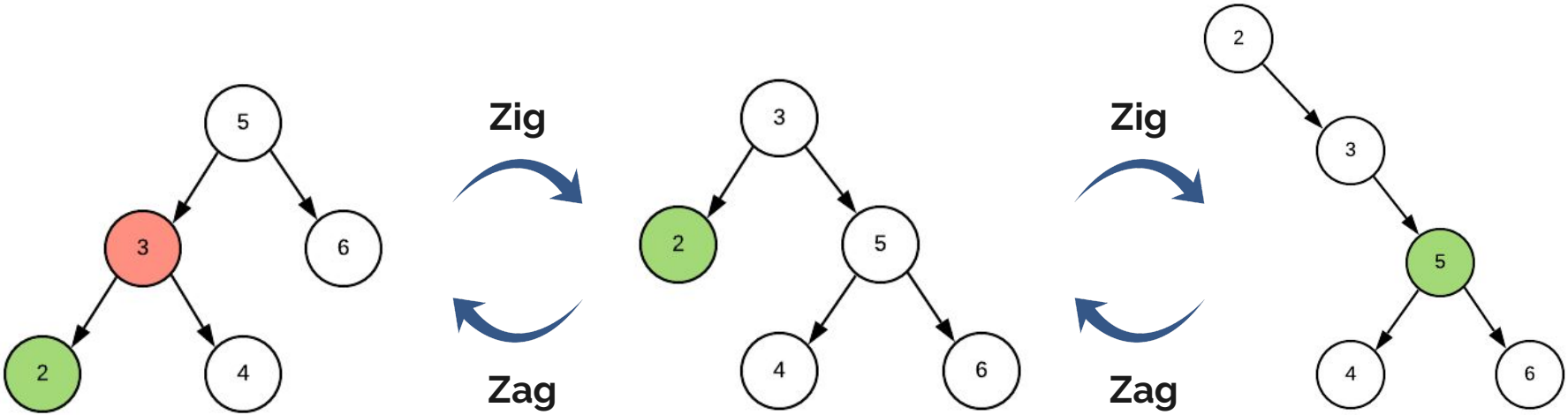
Zig Rotation

Rotate the node one position to the right from its current position:



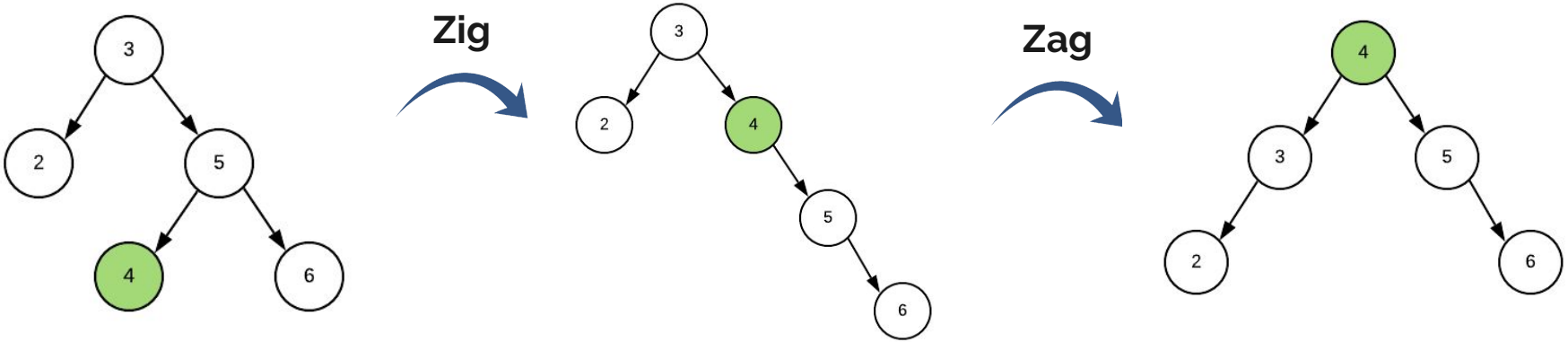
Zig-Zig Rotation

Rotate the node two positions to the right from its current position:



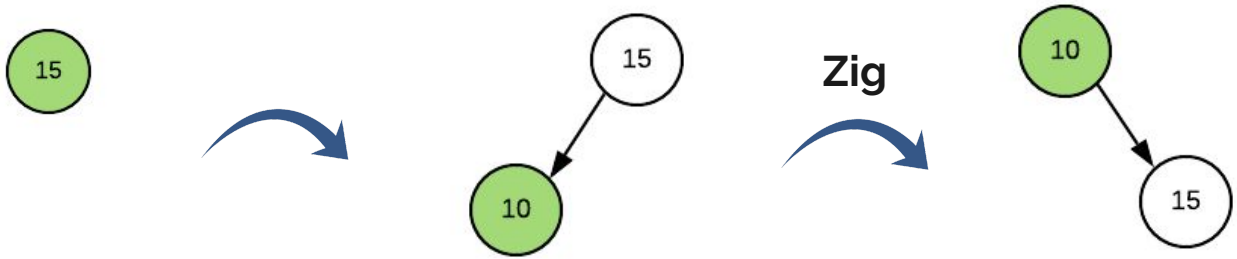
Zig-Zag Rotation

Rotate the node one position to the right followed by one position to the left from its current position:



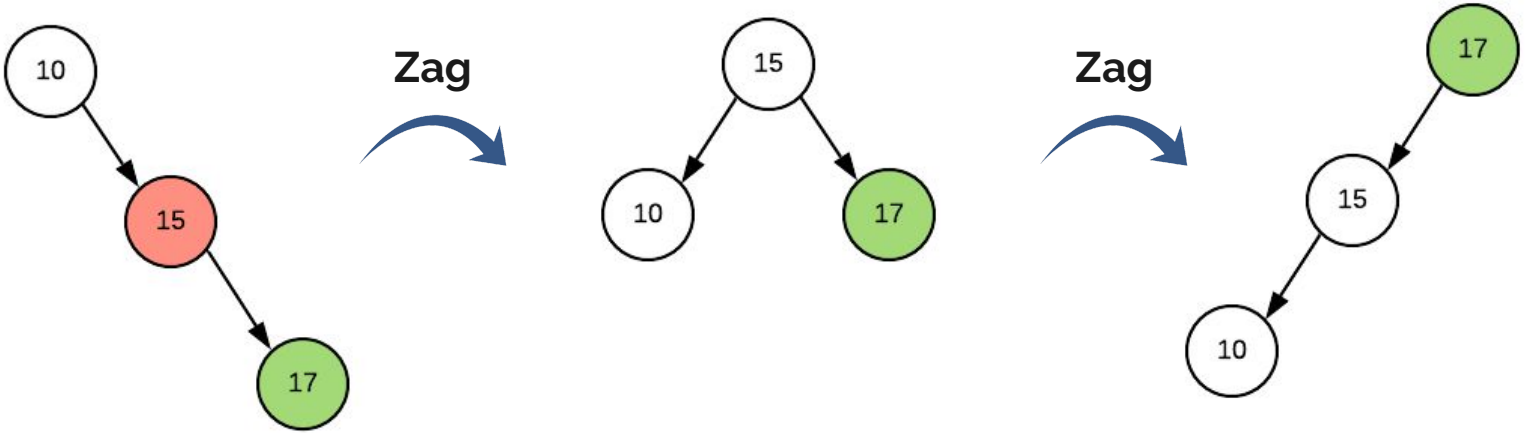
Insert

15, 10, 17, 7



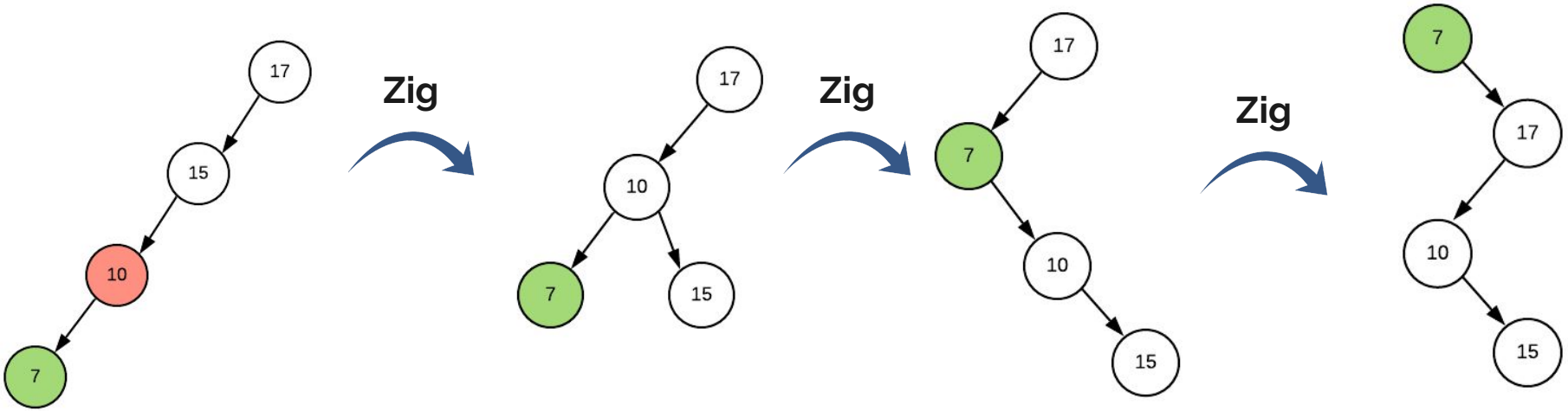
Insert

15, 10, 17, 7



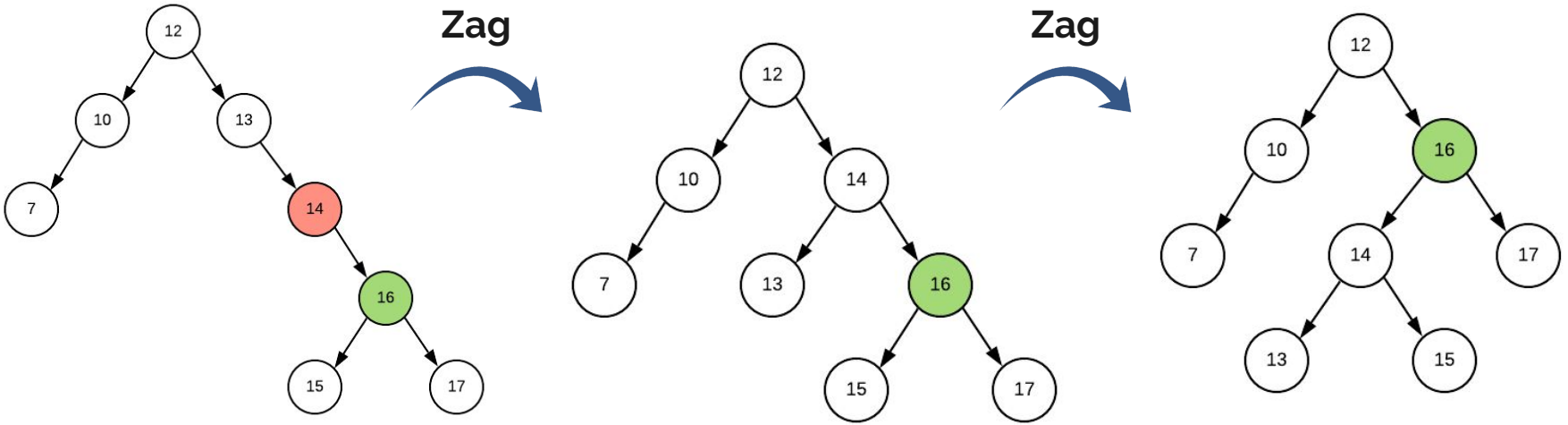
Insert

15, 10, 17, 7



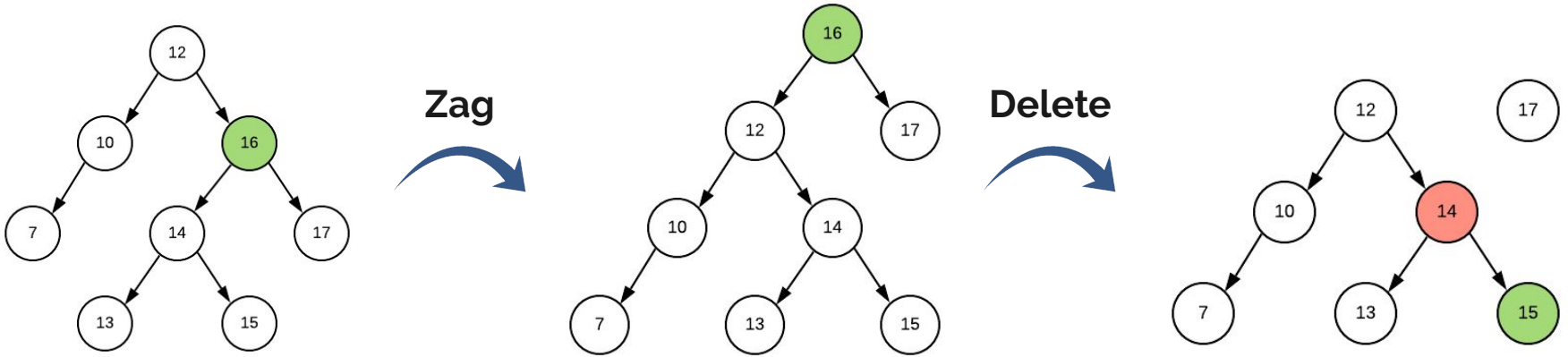
Delete

16


12
10
13
7
14
16
15
17
Zag
12
10
14
7
13
16
15
17
Zag
12
10
16
7
14
17
13
15

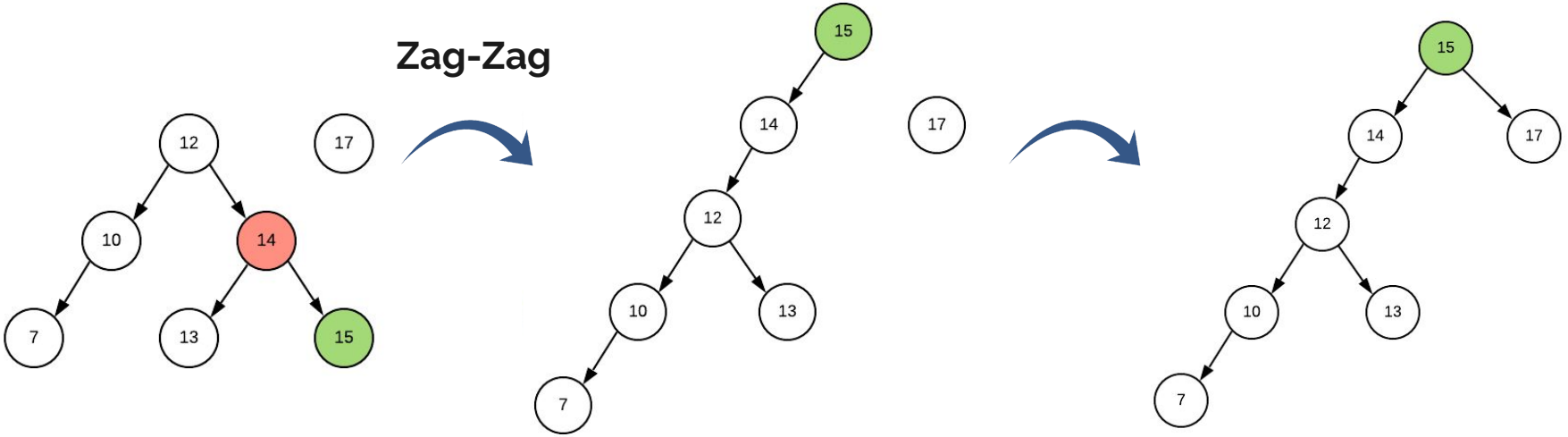
Delete

16


12
10
16
7
14
17
13
15
Zag
16
12
17
10
14
7
13
15
Delete
12
17
10
14
7
13
15

Delete

16



Complexity

Space: O(n)

Time: O(log(n)) [Worst: O(n)]

Applications

- Caches
- Network Router
- Data compression
- Garbage collection

Questions

1. Is doing a Zig-Zig rotation the same as doing 2 Zig rotations?
2. Why would we require to move the node that we are going to delete to be the root?

Thank You!