Balanced Trees

Splay trees

2-3-4 trees

Red-black trees

B-trees

Reference: Chapter 13, Algorithms in Java, 3rd Edition, Robert Sedgewick.

Symbol Table Review

Symbol table: key-value pair abstraction.

- Insert a value with specified key.
- Search for value given key.
- Delete value with given key.

Randomized BST.

- log N time per op (unless you get ridiculously unlucky).
- Store subtree count in each node.
- Generate random numbers for each insert/delete op.

This lecture.

- Splay trees.
- 2-3-4 trees.
- Red-black trees.
- B-trees.

Splay Trees

Splay trees = self-adjusting BST.

- Tree automatically reorganizes itself after each op.
- After inserting x or searching for x, rotate x up to root using double rotations.
- Tree remains "balanced" without explicitly storing any balance information.

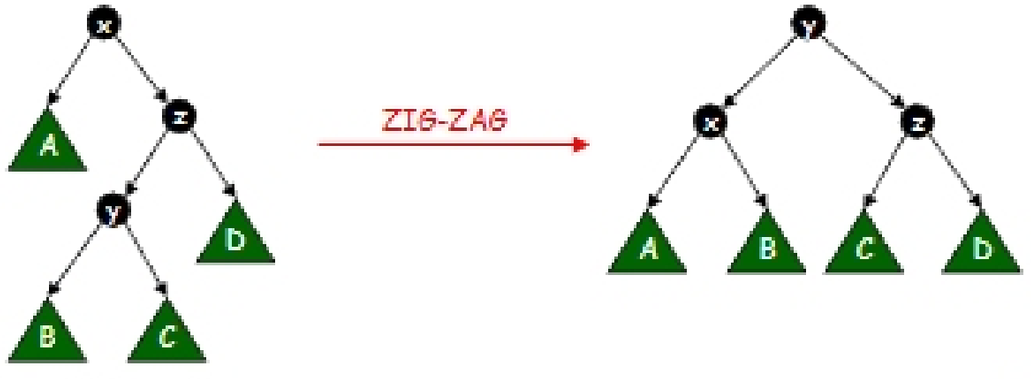
Amortized guarantee: any sequence of N ops takes O(N log N) time.

- Height of tree can be N.
- Individual op can take linear time.

Splay Trees

Splay.

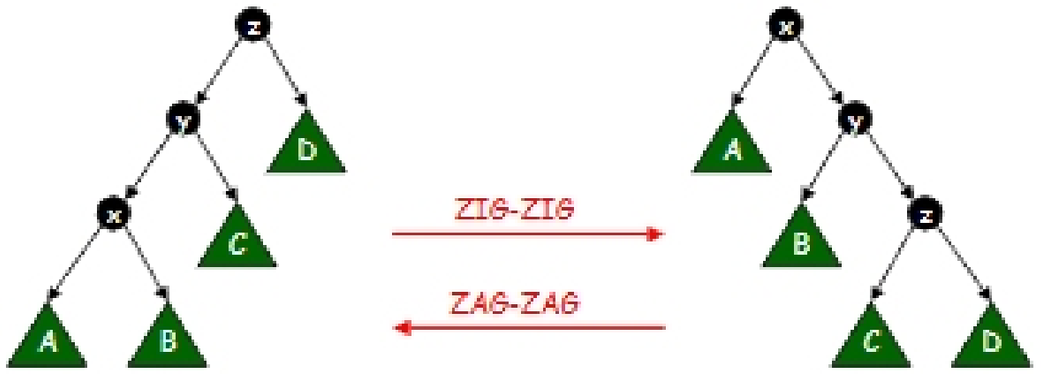
- Check two links above current node.
- ZIG-ZAG: if orientations differ, same as root insertion.
- ZIG-ZIG: if orientations match, do top rotation first.



Splay Trees

Splay.

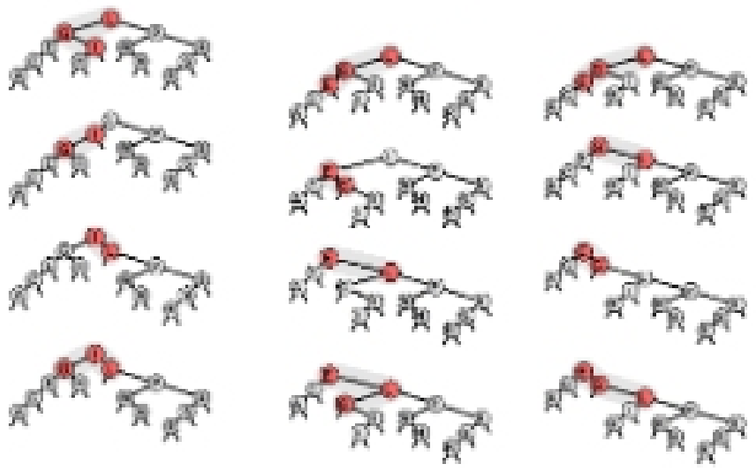
- Check two links above current node.
- ZIG-ZAG: if orientations differ, same as root insertion.
- ZIG-ZIG: if orientations match, do top rotation first.



Splay Trees

Splay.

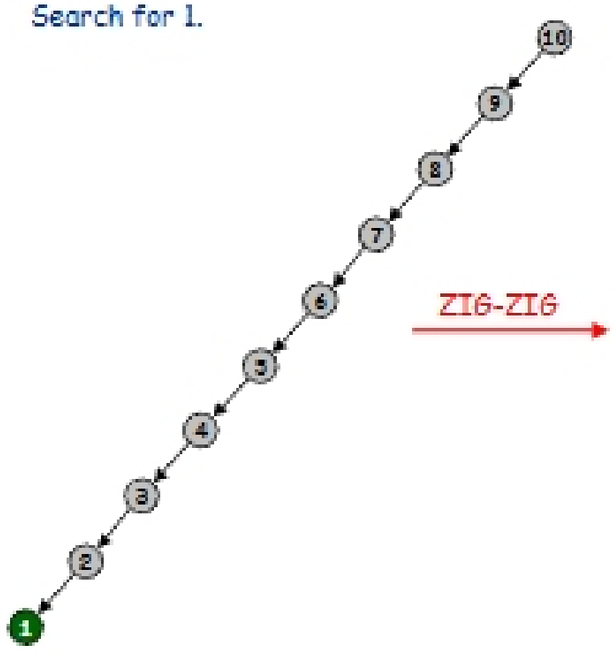
- Check two links above current node.
- ZIG-ZAG: if orientations differ, same as root insertion.
- ZIG-ZIG: if orientations match, do top rotation first.



Root = Splay Root Insertion Splay Insertion

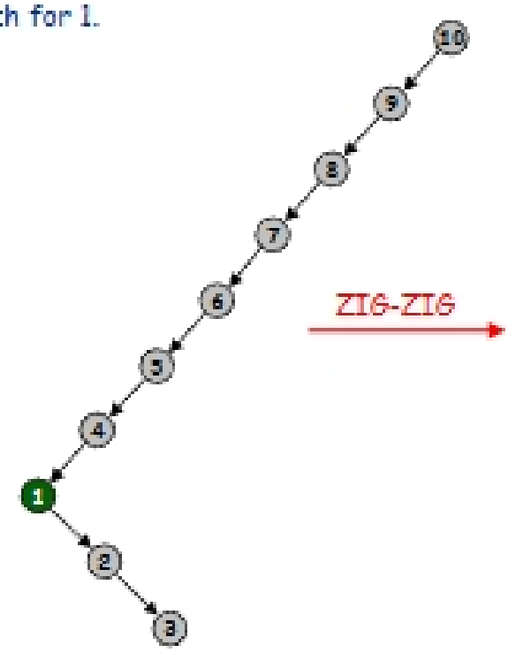
Splay Example

Search for 1.



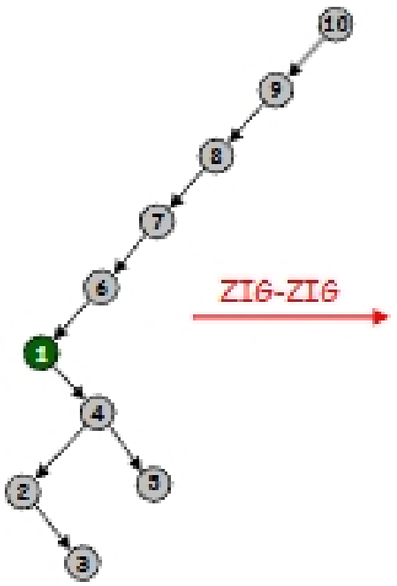
Splay Example

Search for 1.



Splay Example

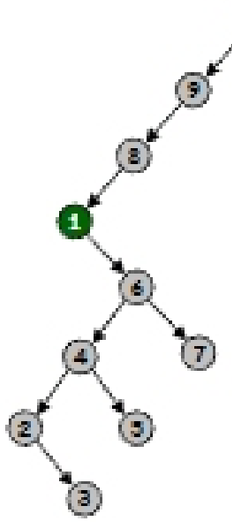
Search for 1.



ZIG-ZIG

Splay Example

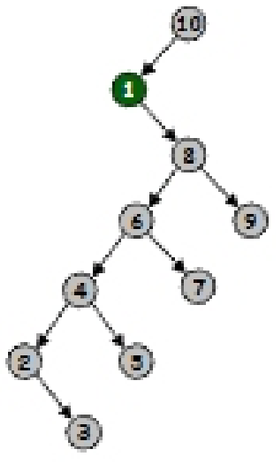
Search for 1.



ZIG-ZIG

Splay Example

Search for 1.



ZIG

Splay Example

Search for 1.