**Basic idea** -- after each time we access a node (insert, search), it gets pushed upward to the root by a series of AVL rotations.

i.e. (Does this tree contain xx) → splay tree

Every time you do an insert, delete, or contains(boolean), the tree changes.

**Rotation in Splay Trees**

*Assume node X is the one that needs to be splayed to the root*

* (zig) Node X has a parent, *P*, and no grandparent
  + This means X is almost the tree root… one level down
  + Use normal AVL rotation
* (zig-zig) Node X is R-Child of Parent, *P*, and *P* is R-Child of Grandparent G
* (zig-zag) Node X is R-Child of Parent*, P*, and *P* is L-Child of Grandparent G

**Find Min/Max**

When you find min/find max, you move the tree all to one side. The node that is the min/max becomes the new root

**Remove**

Remove node in the same way as BST but make the parent of the removed node the new root

**Remove** (Alternate)

Splay node be moved to root

Remove root leaving 2 disconnected subtrees

Join subtrees

SPLT Complexity

* Splays lengthed the path for some nodes
* You cannot guarantee O(log N) bound for any single operation
* O(N) worst case still holds for insert, find, etc.