Balanced Trees

We consider trees that can support

MEMBER
INSERT
DELETE
MIN
MAX
JOIN
SPLIT

etc.

in $O(\log(n))$ worst case time per operation.
Idea:

• Limit the set of possible tree shapes to ones that have height $O(\log(n))$, but with enough flexibility to make "adjustment" of the tree relatively easy after an operation disturbs its balance.

• We have already seen how binary search trees can support a sequence of $n$ operations in $O(n\log(n))$ expected time, and how self-adjusting binary search trees could improve this bound to $O(n\log(n))$ in the worst case by walking back up to the root to "adjust" the tree after each operation (by "juggling" vertices via left and right rotations).

• All of our approaches here will be in the same spirit, except that an extra bit will be stored and maintained with each vertex that will allow the "juggling" by rotations to have the more precise effect of $O(\log(n))$ worst case time per operation.