CSE 332: Data Structures and Parallelism

Section 4: Balanced Trees Solutions

0. The *A***BC**'s of *A***VL** Trees

What are the constraints on the data types you can store in an AVL tree? When is an AVL tree preferred over another dictionary implementation, such as a HashMap?

Solution:

AVL trees are similar to TreeMaps. They require that keys be orderable, though not necessarily hashable. The value type can be anything, just like any other dictionary.

A perk over HashMaps is that keys are stored and can be iterated over in sorted order.

1. Let's Plant an AVL Tree.

Insert 10, 4, 5, 8, 9, 6, 11, 3, 2, 1, 14 into an initially empty AVL Tree. **Solution:**



2. MinVL Trees

Draw an AVL tree of height 4 that contains the minimum possible number of nodes. **Solution:**



3. AVL Trees

Insert 6, 5, 4, 3, 2, 1, 10, 9, 8, 7 into an initially empty AVL Tree.

Solution:

