CSE 332: Data Abstractions

Section 4: Balanced Trees

0. MinVL Trees

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

1. AVL Trees

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

2. AVL Trees

Given a binary search tree, describe how you could convert it into an AVL tree with worst-case time $\mathcal{O}(n \lg(n))$. What is the best case runtime of your algorithm?

3. HeapVL Trees

Is there an AVL Tree that isn't a heap? Is there a heap that isn't an AVL tree? Is there a binary search tree that is neither? Is there a binary search tree that is both?

4. B-Trees

(a) Insert the following into an empty B-Tree with $M = 3$ and $L = 3$: $12, 24, 36, 17, 18, 5, 22, 20$.

(b) Delete $17, 12, 22, 5, 36$

(c) Given the following parameters for a B-Tree with $M = 11$ and $L = 8$

- Key Size = 10 bytes
- Pointer Size = 2 bytes
- Data Size = 16 bytes per record (includes the key)

Assuming that M and L were chosen appropriately, what is the likely page size on the machine where this implementation will be deployed? Give a numeric answer and a short justification based on two equations using the parameter values above.