AVL Trees Vs. Heaps

Can we use an AVL tree to sort?

(A thought question.)

Building AVL Tree: $O(n \log n)$
In-Order Traversal: $O(n)$

- Huh? It takes $O(n)$ to build a heap, but $O(n \log n)$ to build an AVL tree. Why?

- What is the key difference between Heap + AVL?

We care about relative ordering of siblings.

We care about hierarchical ordering. (Relative ordering of siblings doesn't matter!!)

AVL. Heap.
If we built a decision tree to describe BuildAVLTree, it would have leaves that were all possible permutations of the input.

How many leaves?

\[ 3! = 3 \times 2 \times 1 = 6 \]

- There are \( n! \) leaf nodes.
- The path length is \( \log(n!) \).
- So the lower bound for work to build an AVL tree is \( O(\log n!) \).
- We learned that this \( O(n \log n) \).
- So AVL Search is \( O(n \log n + n) = O(n \log n) \).

\[ \text{Build AVL} \quad \text{Inorder traversal} \]