159.172 Computational Thinking

Chapter 9: Search Trees

Searching is a very common operation and therefore has been discussed extensively in this course. Linear search of a list is slow, but it can be improved with a binary search. However, although search time can be improved, insertion and deleting from a list can still be time consuming, since the order must be preserved. Using a tree structure can make search, insertion, and deletion more efficient.

Search tree contain nodes that have keys (sometimes called payloads) associated with their data. The nodes in the search tree are organised based on the relationship between keys.

9.1 Binary Search Trees

Back to top

A binary search tree is a binary tree in which each node contains a search key and the tree is structured such that for each interior node V the following holds:

- All keys less than the key in node V are stored in the left subtree of V.
- All keys greater than the key in node V are stored in the right subtree of V.

Searching for a key in a binary search tree, inserting a new node, and deleting a node are all $O(n)$ operations, where n is the number of nodes. This is because we have to assume that in the worst case the height of the tree is n, and the height determines the complexity of the operations.

9.2 AVL Trees

Back to top

AVL trees are balanced binary trees, which means that their height is as small as possible (with some small margin of error). This guarantees that operations like searching for a key, inserting a node, and deleting a node are linear in the height of the tree, which means they are $O(\log n)$.

Since inserting or deleting nodes can cause the tree to become unbalanced, we need to rebalance the tree after performing these operations. This requires extra effort, but inserting and deleting nodes is still $O(\log n)$.

9.3 2-3 Trees

Back to top

A 2-3 tree is a search tree that is always balanced and whose shape and structure is defined as follows:

- Every node has capacity for one or two keys (and their corresponding payload).
- Every node has capacity for up to three children.
- All leaf nodes are at the same level.
- Every internal node must contain two or three children. If the node has one key, it must contain two children; if it has two keys, it must contain three children.

The 2-3 tree provides fast operations ($O(\log n)$) for insertion and deletion, which are easy to implement (unlike the operations for an AVL trees).

Back to top