Balanced

Binary Search Trees - lexicographic

\[
\begin{align*}
\text{FIND (SEARCH)} & \quad \Theta(\text{height}) \\
\text{INSERT/DELETE} & \\
\text{PRED/SUCCE} & \\
\end{align*}
\]

\(\text{height} \leq \Theta(n)\)

- AVL (height-balanced) \implies \text{Fibonacci recurrence}
- BB (height-balanced) \implies \text{geometric application}
- 2-3 trees (3-2 trees) \implies \beta\text{-trees (external memory)}

Red/Blacks Trees (Binary)

Binary trees
- nodes are either red or black
- leaves are black
- root is black
- no parent/child red nodes

Every leaf has the same number of black nodes on the path from it to the root
2-3-4 trees

Internal node

All leaves are at the same depth from the root

\( \text{height of a 2-3-4 tree} = \Theta(\log n) \)

Insertion
\( \Theta(\log n) \)

Deletion
\( \Theta(\log n) \)