contains \( \mathcal{P}(x) \):

\[
\text{Time} = O(\text{height} +) 
\]

\[
\begin{array}{c}
3 \\
<3 \\
\end{array}
\]

\[
\begin{array}{c}
>3 \\
\end{array}
\]
On average, height is $O(\log n)$.

Inserting $\{1, 2, \ldots, n\}$ in order:

1. $O(\log n)$ time

For $i$ insert $i$

Total of $\sum_{i=1}^{n} O(i) = O(n^2)$
Need a balance condition that guarantees height is $O(\log n)$.

Trees with a balance condition called balanced binary search trees.
Idea 1:
Force root to have two subtrees of same height.
Idea 2: Every node has two subtrees of same height.
Can only hold for perfectly balanced binary search trees. Height $h$ implies exactly $2^h - 1$ nodes.
Adelson-Veski and Landis [62]: For every node, height of subtrees differ by at most 1.

(Use convention that empty subtrees have height -1.)
If you have this condition, it's an AVL tree.

\[ S(h) = \max \# \text{nodes in an AVL tree of height } h. \]