**Binary Search Trees**

- \( \Theta(\log n) \) for search
- \( \Theta(n) \) for insertion
- \( \Theta(n) \) for deletion

**Dynamic**
- \( \Theta(\log n) \) for search in balanced trees

**Balanced Trees**

- \( \Theta(1) \)
- \( \Omega(1) \)

**Search Time**

- \( \text{height} = \text{search time} \)
What is average? — Set of all trees of \( n \) nodes.

Root of a given tree:

Expected height? \[ \frac{1}{5} \frac{1}{5} \frac{1}{5} \frac{1}{5} \frac{1}{5} \]
"uniform forest of trees"

expected height

AVG of \( \text{MAX} \)

\[
\text{AVG search time} = \sum_{t} p(t) \text{avg}(t)
\]

\[
L(t) = \sum_{t} p(t) L(t) = c \log n
\]

\[
\frac{c > 2}{c \log \frac{2e}{c} = 1}
\]

\( \Rightarrow c \approx 4.3 \)

What is avg search time in BST?

\[
\text{EPL}(T) = \sum \text{depth}(t)
\]

\[
\text{IPL}(0) = 0
\]

\[
\text{EPL}(T_r) = \text{EPL}(T_c) + \text{EPL}(T_r) + n+1
\]

\[
\text{EPL}(T) = 2^h = 2n
\]

\[
0 \quad \text{IPL}(0)
\]

\[
\text{IPL}(T) = \text{IPL}(T_c) + \text{IPL}(T_r) + n-1
\]

n internal nodes

n+1 leaves

n internal nodes

(n+1 leaves)
Expected EPL (t)

\[ E_n = \begin{cases} 0 & \text{if } h = 0 \\ \frac{1}{n} \sum_{h=1}^{n} (E_{h-1} + E_{n-h} + n+1) \end{cases} \Pr(h_{\text{largest}}) \]

\[ = \frac{1}{n} \sum_{h=1}^{n} (E_{h-1} + E_{n-h} + n+1) \]

\[ = n+1 + \frac{1}{n} \begin{pmatrix} E_0 + E_{n-1} \\ E_1 + E_{n-2} \\ \vdots \\ E_{n-1} + E_0 \end{pmatrix} \]

\[ E_n = n+1 + \frac{2}{n} \sum_{h=0}^{n} E_h \]

\[ = 2n \ln n \]

\[ = \frac{2n \ln n - cn}{n+1} \Rightarrow \ln n = \Theta(\log n) \]

\[ I_n = E_n - 2n \]