- Lab closed
- Final exam: Wed. Dec. 17, 8-11 am

A BST is a BST.

Convention: nil children are considered to be leaves.

**RBT Properties**

1. Each node Red or Black.
2. Root is Black
3. Each leaf (i.e. nil) is Black.
4. Each Red node has 2 Black children
5.) for any node $x$, every descending path from $x$ to a leaf has same # of black nodes.

Define **the black height** $bh(x)$, of $x$ is the # of black nodes in a tree. Path from $x$ to a leaf (not counting $x$ itself), well defined by **BST** #5.
<table>
<thead>
<tr>
<th>nodes</th>
<th>bluee height</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>5, 11, 3</td>
<td>2</td>
</tr>
<tr>
<td>1, 2, 4, 6, 7, 9, 10, 12</td>
<td>1</td>
</tr>
<tr>
<td>all leaves (niles)</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note**: \( bh(x) = 0 \) if \( \text{height}(x) \) = 0.

1. \( x \) is a leaf.
**Theorem**

A RBT with \( n \) internal (non-nil) nodes and height \( h \) satisfies

\[
h \leq 2 \log (n+1)
\]

**Proof**

Any binary tree satisfies \( h \geq \lceil \log n \rceil \), so a RBT \( T \) satisfies

\[
\Omega(\log n) \leq \text{height}(T) \leq O(\log n)
\]

so \( \text{height}(T) = \Theta(\log n) \).

Thus all BST algorithms run in time \( \Theta(\text{height}(T)) = \Theta(\log n) \).
A Binary Tree satisfying

\[ \text{height}(T) = \Omega(\log n) \]

is often called a balanced tree.

- Insert, Delete, Search for

  BSTs do not preserve RB-T

  properties. However, they can
  be altered so as to preserve
  RB-T properties, and to run in
  time \( \Theta(\text{height}(T)) = \Theta(\log n) \).
If $x$ is any node in a RB tree, then the subtree rooted at $x$ contains at least \( \frac{bh(x)}{2} - 1 \) internal nodes.

**Notation**

\[ N(x) = \# \text{ of internal nodes in the subtree rooted at } x \]

**Lemma says**:

\[ N(x) \geq 2^{\frac{bh(x)}{2}} - 1 \]
\textbf{Proof.} (Induction on } \text{height}(x)\).

If } \text{height}(x) = 0\), then } x \text{ is a leaf, and } \text{bh}(x) = 0\), so

\[ 2^{\text{bh}(x)} - 1 = 2^0 - 1 = 1 - 1 = 0 \]

since } x \text{ is a leaf, } N(x) = 0\), so
inequality is } 0 \geq 0\), which is true.

Let } \text{height}(x) > 0\) and assume for any node } y \text{ with } \text{height}(y) < \text{height}(x)\) that

\[ N(y) \geq 2^{\text{bh}(y)} - 1 \]

we must show:

\[ N(x) \geq 2^{\text{bh}(x)} - 1 \]
Since $\text{height}(x) > 0$, $x$ is an internal node, and necessarily has 2 children (one or both may be null).

1. If $\text{color}(x) = \text{Red}$, then
   \[ bh(\text{left}(x)) = bh(x) \]

2. If $\text{color}(x) = \text{Black}$, then
   \[ bh(\text{left}(x)) = bh(x) - 1 \]

In any case we have

\( \bigcirc \) \hspace{1cm} bh(\text{left}(x)) \geq bh(x) - 1

Since $\text{height}(\text{left}(x)) < \text{height}(x)$, the (strong) ind. hyp. gives
\[ N(\text{left} \times 1) \geq 2 \quad \text{by ind. hy.} \quad \text{by } \circ \]

a similar argument for right \times 1

gives

\[ N(\text{right} \times 1) \geq 2 \quad \text{by ind. hy.} \quad \text{by } \circ \]

\[ \text{analogous inequality to } \ast \]

\[ N(x) = N(\text{left} \times 1) + N(\text{right} \times 1) + 1 \]

\[ \geq (2 \cdot bh(x) - 1) + (2 \cdot bh(x) - 1) + x \]

\[ = 2 \cdot 2 \cdot bh(x) - 1 \]

\[ = 2 \cdot bh(x) - 1 \]
Proof of Theorem:

Let $T$ be a $2B$-tree with $n$ internal (key-bearing) nodes, let $h = \text{height}(T) = \text{height}(\text{root}(T))$.

By $2B$-Tree 4, at least half the nodes in any desc. path from root to leaf must be black (otherwise would have 2 reds in a row.) Thus

$$bh(\text{root}(T)) \geq \frac{h}{2}$$
By previous lemma:

\[ n = N\left(\text{root}[T]\right) \geq 2^{\left\lfloor \frac{h}{2} \right\rfloor} - 1 \geq 2^{-1} \]

\[ n \geq 2^{-1} \]

\[ 2^{-1} \leq n \]

\[ 2^{-1} \leq n+1 \]

\[ \frac{h}{2} \leq \log(n+1) \]

\[ h \leq 2 \log(n+1) \]

\[ \therefore \]
13.2 Rotation

\[
\text{RightRotate}(T, y)
\]

\[
\text{LeftRotate}(T, x)
\]