Theorem (07HeapSort Lecture) A complete binary tree with \( n \) nodes has height \( h = \lceil \log_2(n) \rceil \).

Proof: A complete binary tree of height \( h \) can accommodate between \( 2^h \) and \( 2^{h+1} - 1 \) nodes. If \( n \) is the number of nodes, let \( h \) be the smallest number for which

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

The left-hand inequality yields \( h \leq \log_2 n \) which in turn implies that \( h = \lceil \log_2 n \rceil \).

Theorem: A red-black tree with \( n \) internal nodes has height at most \( h = \lceil 2 \log_2(n+1) \rceil \).

NOTE: This implies that in the worst-case scenario, a red-black tree is (approximately) only twice as deep as the corresponding complete (or perfect) binary tree. To prove this theorem, we need a lemma:

Lemma: For any node \( x \), the subtree rooted at \( x \) contains at least \( 2^{bh(x)} - 1 \) internal nodes (i.e. nodes that have children).

Proof: Proceed by induction on the height \( h(x) \) of the tree whose root is the node \( x \). If the height if \( h(x) = 0 \), then the node must be a leaf (remember the height is measured from the bottom of the tree and the depth is measured from the top (root) of the tree) which must necessarily have black-height of 0. Furthermore, this leaf node also contains at least \( 2^{bh(x)} - 1 \) = \( 2^0 - 1 \) = 0 nodes, so the basis step of the induction is true.

Assume now that the result holds for any subtree rooted at a node of positive height \( n \), and let \( x \) be a node whose height is \( n + 1 \). We need to prove that result holds for this node \( x \). To do this, consider each of the children of node \( x \), both of whose heights must be \( n \) so the inductive hypothesis will apply. Notice that the black-height of each of the child nodes with either be equal to the black-height of the parent \( bh(x) \) or one less than the black-height of the parent \( bh(x) - 1 \) depending on whether that child is red or black, respectively. Thus, in the worst-case scenario in which both of the children are black, by the inductive hypothesis we known that each child would have at least \( 2^{bh(x)} - 1 \) = \( 2^0 - 1 \) = 0 nodes, so the total number of nodes in the tree rooted at \( x \) must be at least the sum of the two worst-case scenarios for the children plus one for the root node \( x \):

\[
2^{bh(x)} - 1 + 2^{bh(x)} - 1 - 1 + 1 = 2 \cdot 2^{bh(x)} - 1 = 2^{bh(x)} - 1.
\]

This proves the result is true for a node of height \( n + 1 \), so the lemma is true by induction.

Proof of Theorem: Suppose that \( h \) is the height of the given red-black tree with \( n \) nodes. Because any red parent within a red-black tree must necessarily have two black children, we known that any path from the root to a leaf must necessarily contain at least \( h/2 \) black nodes. In particular, if \( x \) is the root node, then \( bh(x) \geq h/2 \), so by our lemma the corresponding tree must contain at least

\[
2^{bh(x)} - 1 \geq 2^{h/2} - 1
\]

internal nodes \( n \), thus \( n \geq 2^{h/2} - 1 \). Solving this expression for \( h \) gives \( \log_2(n+1) \geq h/2 \) from which it follows that \( h \leq \lceil 2 \log_2(n+1) \rceil \).