* Full Binary Tree
* Complete Binary Tree
* Height-Balanced Tree
* Full Binary Tree Theorem:
  + The number of leaves in a non-empty Full Binary Tree is one more than the number of internal nodes.
  + Proof (math induction)
    - Base case: A Full Binary Tree with 0 internal nodes has one leaf node.
    - Induction Hypothesis:
      * Assume any Full Binary Tree T containing n-1 internal nodes has n leaves.
    - Induction step:
      * Given a Full Binary Tree T with n-1 internal nodes, there should be n leaves.
      * Add two leaf nodes as children of one of its leaves, then we get tree T’
* Theorem 2:
  + The number of empty subtrees (null links) in a non-empty Binary Tree is one more than the number of nodes in the tree.
  + Applies to any Binary Tree structure.
  + Tells us that the number of null links in a tree linked implementation takes up about half of the memory space.
* Array implementation for Complete Binary Tree
  + Level-by-level storage of the complete Binary Tree (breadth-first traversal).
  + Parent (r) = [(r-1)/2, if r != 0 and r<n]
    - N is the number of nodes in the tree, r is an index in the array
  + LeftChild (r) = [2r+1, if 2r+1<n]
  + RightChild (r) = [2r+2, if 2r+2<n]
  + LeftSibling (r) = [r-1, if r is even, r>0, and r<n]
  + RightSibling (r) = [r+1, if r is odd and r+1<n]
  + How to determine if index i corresponds to a leaf node?
    - If i >= n/2, then true