HINTS:

1) Think about what end of the Deque you used for addFirst(), addLast(), removeFirst(), removeLast(), etc. on the homework.
2) Pay attention to where front and back should always be pointing to, especially during wraparound.
3) height = max{node.left.height, node.right.height} + 1 (number of edges from the node to its deepest descendant)

   depth of a node is the number of edges from the root to the node

   BST has two characteristics: It is a binary tree, and it has the order property (left child < node < right child's data).

   Complete Trees: Every level is filled except the last level, which is filled from left to right.

   Leaf: Node without children

4) Take a look at addToBack on your DoublyLinkedList homework. How would you replace each time you use the tail?
5) Most of these are slight modifications of homework methods.
6) Think about what nodes are going to be accessed in the list, and what qualifies a node to be added to the output list.
7) This is a slight modification of get().