QuadTree Visualization

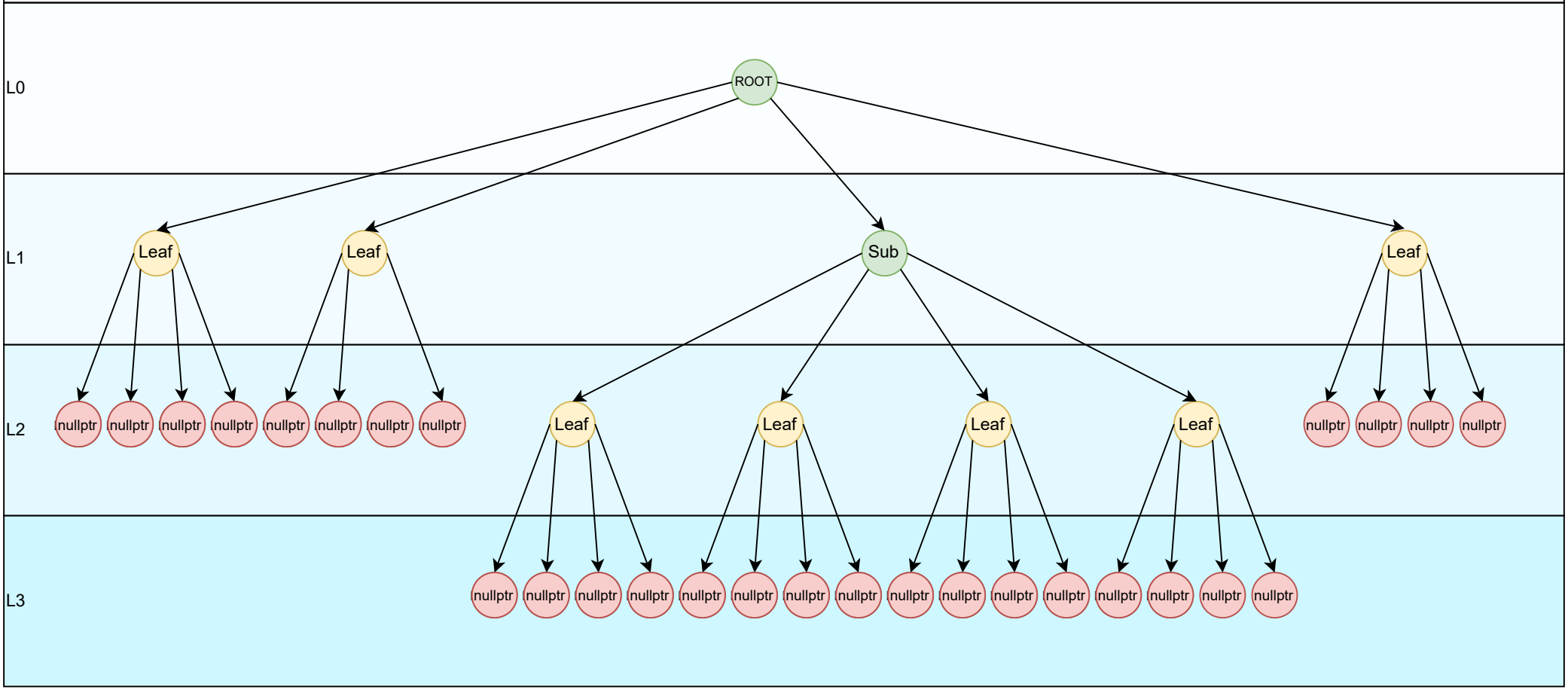
KEY:

Full

Leaf

NULL

- Each node represents a QuadTree object
- The root node is allocated on the stack when the simulation starts
- All other nodes are dynamically allocated on the heap and need to have memory managed
- Each node has an array of pointers to QuadTree objects of size 4
- Each array of QuadTree pointers is either fully filled with dynamically allocated QuadTree objects or each index in the array points to nullptr
- Each frame when QuadTree::insert() is called the tree is built and nodes are inserted and split as necessary
    - Each node will have 0 to MAX_CAPACITY particles within the its region
    - Split happens when particles in node surpasses MAX_CAPACITY
    - MAX_CAPACITY of 1 means a split occurs when 2 nodes are in the QuadTree
    - If there is a split, reinsert particles from parent node into appropriate subtrees
- QuadTree::deleteTree() is called before particles are inserted and should recursively free all nodes up to the root node



Example of Particle positions that would generate the above QuadTree with MAX_CAPACITY at 1: