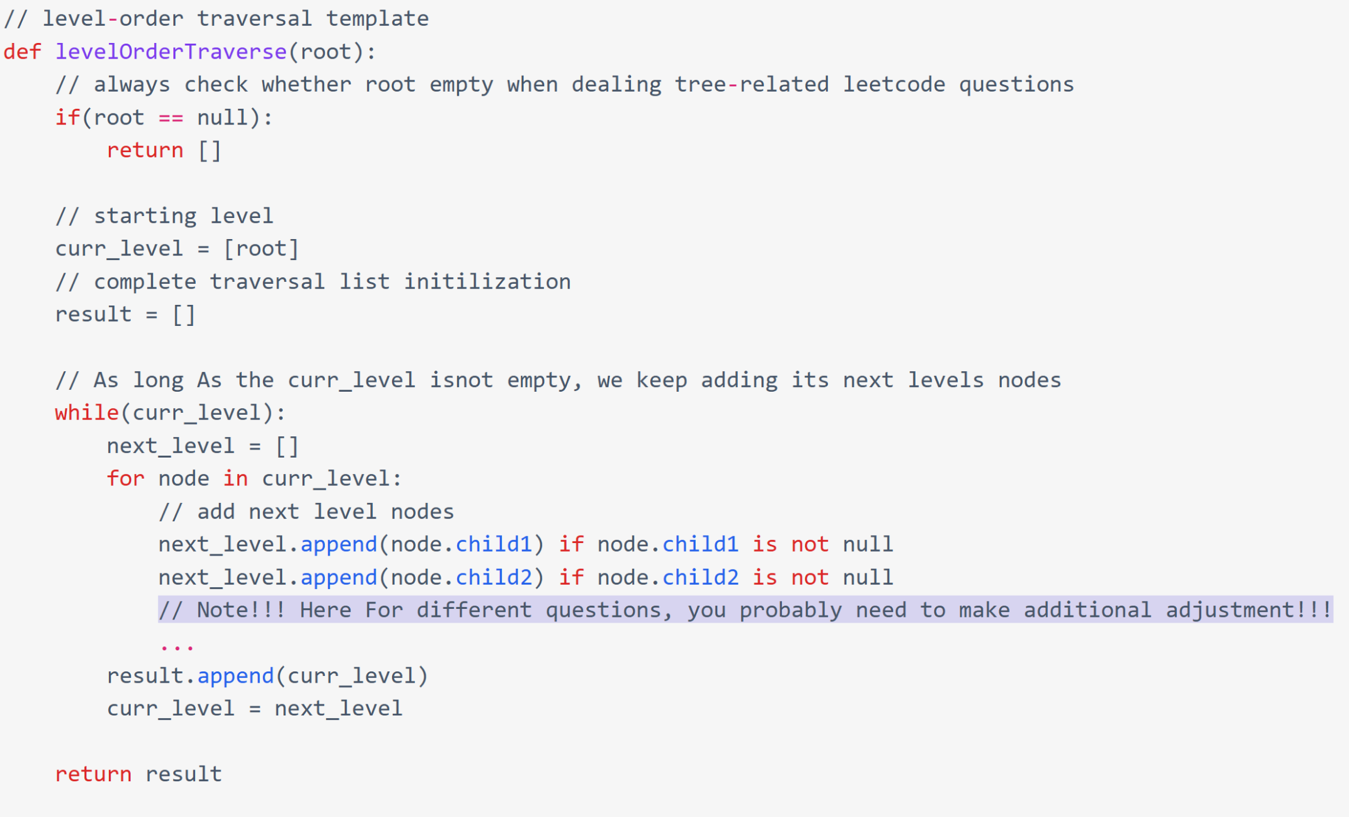
**Recursion & Stack**

**Differences between Recursion and Stack**

Recursion is an automated mechanism built on the language runtime’s call stack: each function invocation implicitly creates an activation record on that stack. In contrast, an explicit stack is a manually constructed LIFO data structure in your code, where you must explicitly call push and pop to save and restore algorithmic state. In other words, recursion is automatic, whereas an explicit stack must be managed by the programmer.

**Pseudocode for level-order traversal(using stack)**

If we need to change the order, we should reverse the values list, rather than appending nodes to next\_level in reverse.



[LeetCode 429]

如果不是tree那样left和right而是children，那么values要用extend而不是append: next\_level.extend(node.children). **extend** 是 Python 列表（list）对象的一个方法，用来把一个可迭代对象（如另一个列表、元组等）中的所有元素，逐个添加到当前列表的末尾。