2024-07-27 - Handout – TREES

What is a Tree ?

Frequently asked Tree implementations :

1. Binary tree

2. Binary search tree

3. Trie

Q1. Binary Tree Level Order Traversal

leetcode link

Given the root of a binary tree, return the level order traversal of its nodes' values. (i.e., from left to right, level by level).

Example 1:
Input: root = [3,9,20,null,null,15,7]
Output: [[3],[9,20],[15,7]]

Example 2:
Input: root = [1]
Output: [[1]]

Example 3:

Input: root = []
Output: []

Q2. Binary Tree inorder traversal

leetcode link

Given the root of a binary tree, return the inorder traversal of its nodes' values.

Example 1:
Input: root = [1,null,2,3]
Output: [1,3,2]

Example 2:
Input: root = []
Output: []

Example 3:

Input: root = [1]
Output: [1]

Q3. Implement Trie (Prefix tree)

leetcode link

A trie (pronounced as "try") or prefix tree is a tree data structure used to efficiently store and retrieve keys in a dataset of strings. There are various applications of this data structure, such as autocomplete and spellchecker.

Implement the Trie class:

Trie() Initializes the trie object.

void insert(String word) Inserts the string word into the trie.

boolean search(String word) Returns true if the string word is in the trie (i.e., was inserted before), and false otherwise.

boolean startsWith(String prefix) Returns true if there is a previously inserted string word that has the prefix prefix, and false otherwise.

Example 1:
Input
["Trie", "insert", "search", "search", "startsWith", "insert", "search"]
[[], ["apple"], ["apple"], ["app"], ["app"], ["app"], ["app"]]
Output
[null, null, true, false, true, null, true]

Q4. Balance a Binary Search Tree

leetcode link

Given the root of a binary search tree, return a balanced binary search tree with the same node values. If there is more than one answer, return any of them.

A binary search tree is balanced if the depth of the two subtrees of every node never differs by more than 1.

Example 1:
Input: root = [1,null,2,null,3,null,4,null,null]
Output: [2,1,3,null,null,null,4]
Explanation: This is not the only correct answer, [3,1,4,null,2] is also correct.

Example 2:
Input: root = [2,1,3]
Output: [2,1,3]

Q4. Lowest Common Ancestor of a Binary Tree II

leetcode link

Given the root of a binary tree, return the lowest common ancestor (LCA) of two given nodes, p and q. If either node p or q does not exist in the tree, return null. All values of the nodes in the tree are unique.

According to the definition of LCA on Wikipedia: "The lowest common ancestor of two nodes p and q in a binary tree T is the lowest node that has both p and q as descendants (where we allow a node to be a descendant of itself)". A descendant of a node x is a node y that is on the path from node x to some leaf node.

Example 1:
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 1

Example 2:
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 4
Output: 5

Output: 3
Explanation: The LCA of nodes 5 and 1 is 3.

Example 3:
Input: root = [3,5,1,6,2,0,8,null,null,7,4], p = 5, q = 10
Output: null
Explanation: Node 10 does not exist in the tree, so return null.

Explanation: The LCA of nodes 5 and 4 is 5. A node can be a descendant of itself according to the definition of LCA.