1. Binary Tree Inorder Traversal

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

Example:

Input: [1,null,2,3]

1

\

2

/

3

Output: [1,3,2]

Follow up: Recursive solution is trivial, could you do it iteratively?

1. Unique Binary Search Trees

Gicen n, how many structurally unique BST’s (binary Search trees) that store values 1 .. n?

Example:

Input : 3

Output : 5

Explanation :

Given n = 3, there are a total of 5 unique BST’s:



1. Symmetric Tree

Given a binary tree, check whether it is a mirror of itself( ie, symmetric around its center)

For example, this binary tree [1,2,2,3,4,4,3] is symmetric:



But the following [1,2,2,null,3,null,3] is not:



Follow up: Solve it both recursively and iteratively.

1. Binary Tree Level Order Traversal

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

For example:

Given binary tree [3,9,20,null,null,15,7],

3

/ \

9 20

/ \

15 7

return its level order traversal as:

[

[3],

[9,20],

[15,7]

]

1. Construct Binary Tree from Preorder and Inorder Traversal

Given preorder and inorder traversal of a tree, construct the binary tree.

Note: You may assume that duplicates do not exist in the tree.

For example,

given preorder = [3,9,20,15,7]

inorder = [9,3,15,20,7]

Return the following binary tree:

3

/ \

9 20

/ \

15 7

1. Binary Tree Zigzag Level Order Traversal

Given a binary tree, return the zigzag level order traversal of its nodes values, (ie, from left to right, then right to left for the next level and alternate between).

For example:

Given binary tree [3,9,20,null,null,15,7].



return its zigzag level order traversal as:

One la Com



1. Convert Sorted List to Binary Search Tree

Given a singly linked list where elements are sorted in ascending order, convert it to a height balanced BST

For this problem, a height balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1.

Example:

Given the sorted linked list: [-10,-3,0,5,9],

One possible answer is [0,-3,9,-10,null,5], which represents the following height balanced BST:



1. Balanced Binary Tree

Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as:

a binary tree in which the left and right subtrees of every node differ in height by no more than 1.

Example 1:

Given the following tree [3,9,20,null,null,15,7]:

3

/ \

9 20

/ \

15 7

Return true.

Example 2:

Given the following tree [1,2,2,3,3,null,null,4,4]:

1

/ \

2 2

/ \

3 3

/ \

4 4

Return false.

1. Flatten Binary Tree to Linked List

Given a binary tree, flatten it to a linked list in-place.



For example, given the following tree

The flattened tree should look like

1



1. Course Schedule

There are a total of numCourses courses you have to take, labeled from 0 to numCourses-1. Some courses may have prerequisites, for example to take course 0 you have to first take course 1, which is expressed as a pair: [0,1]

Given the total number of courses and a list of prerequisite pairs, is it possible for you to finish all courses?

Example 1:

Input: numCourses = 2, prerequisites = [[1,0]]

Output: true

Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0. So it is possible.

Example 2:

Input: numCourses = 2, prerequisites = [[1,0],[0,1]]

Output: false

Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0, and to take course 0 you should also have finished course 1. So it is impossible.

Constraints: The input prerequisites is a graph represented by a list of edges, not adjacency matrices. Read more about how a graph is represented.

You may assume that there are no duplicate edges in the input prerequisites.

1 <= numCourses <= 10^5

1. Binary Tree Preorder Traversal

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

Example:

Input: [1,null,2,3]

1

\

2

/

3

Output: [1,2,3]

Follow up: Recursive solution is trivial, could you do it iteratively?

1. Binary Tree Postorder Traversal

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

Example:

Input: [1,null,2,3]

1

\

2

/

3

Output: [3,2,1]

Follow up: Recursive solution is trivial, could you do it iteratively?

1. Binary Search Tree Iterator

Implement an iterator over a binary search tree (BST).

Your iterator will be initialized with the root node of a BST.

Calling next() will return the next smallest number in the BST.

Example:

BSTIterator iterator = new BSTIterator(root);

iterator.next(); // return 3

iterator.next(); // return 7

iterator.hasNext(); // return true

iterator.next(); // return 9

iterator.hasNext(); // return true

iterator.next(); // return 15

iterator.hasNext(); // return true

iterator.next(); // return 20

iterator.hasNext(); // return false

Note: next() and hasNext() should run in average O(1) time and uses O(h) memory, where h is the height of the tree.

You may assume that next() call will always be valid, that is, there will be at least a next smallest number in the BST when next() is called.