1.1 background introduction

A binary search tree is a [binary tree](https://en.wikipedia.org/wiki/Binary_tree) which may be empty. The tree satisfies the [binary search](https://en.wikipedia.org/wiki/Binary_search) property, which states that the key in each node must be greater than or equal to any key stored in the left sub-tree, and less than or equal to any key stored in the right sub-tree. It support three main operations: insertion of elements, deletion of elements, and lookup (checking whether a key is present). The shape of the binary search tree depends entirely on the order of insertions and deletions, and can become degenerate.

Binary search trees are a fundamental data structure used to construct more abstract data structures such as [sets](https://en.wikipedia.org/wiki/Set_(computer_science)), [multisets](https://en.wikipedia.org/wiki/Set_(computer_science)#Multiset), and [associative arrays](https://en.wikipedia.org/wiki/Associative_array).

1.2 problem description

Given the structure of a binary tree and a sequence of distinct integer keys, we are supposed to find the only way to fill these keys into the tree to make the resulting tree a binary search tree. To present the results, we are asked to output the level order travelsal sequence of the tree. Besides, we shall provide a set of test cases to confirm the accuracy of the program.