Flatten BST to sorted list | Increasing order

Given a binary search tree, the task is to flatten it to a sorted list. Precisely, the value of each node must be lesser than the values of all the nodes at its right, and its left node must be NULL after flattening. We must do it in O(H) extra space where ‘H’ is the height of BST.

Input:
5
/ \
3 7
/ \ / \
2 4 6 8
Output: 2 3 4 5 6 7 8
Input:
1

2

3

4

5
Output: 1 2 3 4 5

Approach: A simple approach will be to recreate the BST from its in-order traversal. This will take O(N) extra space were N is the number of node in BST.
To improve upon that, we will simulate in order traversal of a binary tree as follows:

1. Create a dummy node.
2. Create a variable called ‘prev’ and make it point to the dummy node.
3. Perform in-order traversal and at each step.
    - Set prev -> right = curr
    - Set prev -> left = NULL
    - Set prev = curr

This will improve the space complexity to O(H) in worst case as in-order

traversal takes O(H) extra space.