Convert Binary Search Tree to Max Heap [GFG](https://practice.geeksforgeeks.org/problems/bst-to-max-heap/1)

Given a Binary Search Tree. Convert a given BST into a Special Max Heap with the condition that all the values in the left subtree of a node should be less than all the values in the right subtree of the node. This condition is applied on all the nodes in the so converted Max Heap.

Example:

10

/ \

7 17

/ / \

5 15 19

/ /

3 13

/

1

Output:

19

/ \

7 17

/ / \

5 13 15

/ /

3 10

/

1

**Approach 1: Main function to convert a BST to a Max Heap**

* **Function Purpose:** Convert a given Binary Search Tree (BST) into a Max Heap.
* **Explanation:**
  + The program first performs an inorder traversal of the BST using Morris Traversal and stores the values in the **inorderAns** vector.
  + It then initializes an index pointing to the end of the vector and calls the **convertBSTToMaxHeapHelper** function, which performs a reverse inorder traversal and assigns values from the vector back to the tree.
  + The vector values are assigned to the BST nodes in reverse inorder, effectively converting it to a Max Heap.
* **Time Complexity:** **O(n), where n is the number of nodes in the BST.**
* **Space Complexity: O(n) for the vector storing inorder traversal values.**