Q.1] The pseudo code is divided into 2 parts :

(i.) Sorting the given list of 'n' items L using Merge sort.

(ii.) Doing an inorder traversal of the BST and filling the nodes with values from the sorted list L'.

(A.) Sorting

→ We use the Mergesort algorithm to sort the given list.

Time complexity = $O(n \log(n))$

(B) Inorder Traversal :

→ We assume we have the sorted list L'. Elements of L' are denoted using '[ ]' brackets. So, L'[1] denotes the first element.

Pseudo Code :

```
1   int i = 1 ;   // global variable

2   makeBST (node* p , int* L') {
3       if (p == NULL) {   // null pointer check
4           return ;
5       }
6       makeBST (p → left , L') ;
7       p → data = L'[i] ;
8       if (i != n) { i++ ; } // avoiding overflow
9       makeBST (p → right , L') ;
10  }
```

Time complexity for Inorder traversal

$= O(n)$

Total time complexity of the entire algorithm $= O(n) + O(n \log(n))$

$= O(n \log(n))$

(c) Correctness:

→ The code is similar to Inorder Traversal algorithm. We prove the correctness using Structural Induction.

Base Case 1: When there is only 1 node in BST and length of list is 1 too.

Only line 7 of pseudo code affects the BST. Root node is filled with the only element present.

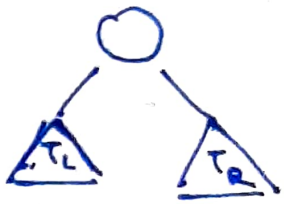
Hence Base case 1 holds true.

Base Case 2: Empty tree

Condition on line 3 holds true. Algorithm outputs nothing.

Hence Base Case 2 holds true.

Constructor case: Assume that the algorithm holds true for left and right subtree of the given tree:



Algorithm starts from root node. Line 6 is executed. Hence T_L is filled first. T_L is correctly filled based on our assumption.

Now, assume T_L has n_L nodes. Hence, value of $i = n_L + 1$ after execution of T_L.

$\because$ L' is sorted,

$L'[1], L'[2] \ldots, L'[n_L] \leq L'[n_L+1]$

Root node is filled with $L'[n_L+1]$.

Hence, value at root node $\geq$ value at any node of T_L

After filling the root node, T_R is filled recursively.

T_R is filled correctly based on our assumption.

$\because$ L' is sorted

$L'[n_L+1] \leq L'[n_L+2], L'[n_L+3] \ldots L'[n]$

Hence, value at root node $\leq$ value at any node of T_R

Hence the entire Bstree is correctly filled.

By structural Induction, the algorithm outputs correctly for any size of BST.