Solution 30-11-23

720. Longest Word in Dictionary

```
import java.util.Arrays;

class TrieNode {
  TrieNode[] children;
  boolean isEndOfWord;

  TrieNode() {
    children = new TrieNode[26];
    isEndOfWord = false;
  }
}

class Trie {
  TrieNode root;

  Trie() {
    root = new TrieNode();
  }

  void insert(String word) {
    TrieNode node = root;
    for (char c : word.toCharArray()) {
      int index = c - 'a';
      if (node.children[index] == null) {
        node.children[index] = new TrieNode();
      }
      node = node.children[index];
    }
```

```
    node.isEndOfWord = true;
  }

  boolean search(String word) {
    TrieNode node = root;
    for (char c : word.toCharArray()) {
      int index = c - 'a';
      if (node.children[index] == null || !node.children[index].isEndOfWord) {
        return false;
      }
      node = node.children[index];
    }
    return true;
  }
}

public class LongestWordInDictionary {
  public static String longestWord(String[] words) {
    Arrays.sort(words);
    Trie trie = new Trie();
    String longestWord = "";

    for (String word : words) {
      if (word.length() == 1 || trie.search(word.substring(0, word.length() - 1))) {
        trie.insert(word);
        if (word.length() > longestWord.length()) {
          longestWord = word;
        }
      }
    }
```

```
    return longestWord;
  }

  public static void main(String[] args) {
    // Example input from the user
    String[] words = { "w", "wo", "wor", "worl", "world" };

    // Output the result
    String result = longestWord(words);
    System.out.println("Longest word: " + result);
  }
}
```

```
PS C:\Users\Ajeet\Desktop\java> javac LongestWordInDictionary.java
PS C:\Users\Ajeet\Desktop\java> java LongestWordInDictionary
Longest word: world
```

897. Increasing Order Search Tree

```
import java.util.Scanner;

class TreeNode {
    int val;
    TreeNode left, right;

    public TreeNode(int x) {
        val = x;
        left = right = null;
    }
}
```

```
public class Solution {
    TreeNode current;

    public TreeNode increasingBST(TreeNode root) {
        TreeNode dummyNode = new TreeNode(0);
        current = dummyNode;
        inOrderTraversal(root);
        return dummyNode.right;
    }

    private void inOrderTraversal(TreeNode node) {
        if (node == null) {
            return;
        }

        inOrderTraversal(node.left);

        // Update pointers for the increasing order
        node.left = null;
        current.right = node;
        current = node;

        inOrderTraversal(node.right);
    }

    public static void main(String[] args) {
        Solution solution = new Solution();
        Scanner scanner = new Scanner(System.in);

        // Example: Let's create a sample binary search tree
        TreeNode root = new TreeNode(5);
```

```
        root.left = new TreeNode(3);
        root.right = new TreeNode(6);
        root.left.left = new TreeNode(2);
        root.left.right = new TreeNode(4);
        root.right.right = new TreeNode(8);
        root.left.left.left = new TreeNode(1);
        root.right.right.left = new TreeNode(7);
        root.right.right.right = new TreeNode(9);

        System.out.println("Original Binary Search Tree:");
        printInOrder(root);

        TreeNode result = solution.increasingBST(root);

        System.out.println("\n\nIncreasing Order Binary Search Tree:");
        printInOrder(result);
    }

    private static void printInOrder(TreeNode node) {
        if (node != null) {
            printInOrder(node.left);
            System.out.print(node.val + " ");
            printInOrder(node.right);
        }
    }
}
```

OUTPUT:-

```
PS C:\Users\Ajeet\Desktop\java> javac Solution.java
PS C:\Users\Ajeet\Desktop\java> java Solution
Original Binary Search Tree:
1 2 3 4 5 6 7 8 9
```

Increasing Order Binary Search Tree:

1 2 3 4 5 6 7 8 9