```java
package trieTree;

/* Trie树的构建、查找 */

import java.util.Arrays;
import java.util.HashSet;

class TrieTree {

    private class TrieNode {

        HashSet<String> stringSet;
        TrieNode[] nextNodes;

        TrieNode() {
            nextNodes = new TrieNode[26];    //只包括小写的a-z
        }
    }

    private TrieNode rootNode;

    public void insertString(String s) {

        if (s == null || s.length() == 0) {
            return;
        }

        s = s.toLowerCase();

        char[] sArray = s.toCharArray();
        Arrays.sort(sArray);

        if (rootNode == null) {
            rootNode = new TrieNode();
        }

        TrieNode tempNode = rootNode;
        for (int i = 0; i < sArray.length; i++) {
            if (tempNode.nextNodes[sArray[i] - 'a'] == null) {
                tempNode.nextNodes[sArray[i] - 'a'] = new TrieNode();
            }
            tempNode = tempNode.nextNodes[sArray[i] - 'a'];

            if (i == sArray.length - 1) {
                if (tempNode.stringSet == null) {
                    tempNode.stringSet = new HashSet<>();
                }
                tempNode.stringSet.add(s);
            }
        }
    }

    public boolean searchString(String s) {

        if ( s == null || s.length() == 0 ) {
            return rootNode == null;
        }

        if ( rootNode == null ) {
            return false;
        }

        s = s.toLowerCase();
        char[] sArray = s.toCharArray();
        Arrays.sort(sArray);
        TrieNode tempNode = rootNode;

        for ( int i = 0; i < sArray.length; i++ ) {
            if (tempNode.nextNodes[sArray[i] - 'a'] == null) {
                return false;
            }
            tempNode = tempNode.nextNodes[sArray[i] - 'a'];
        }
```

```
74
75          return tempNode.stringSet != null && tempNode.stringSet.contains(s);
76      }
77
78      public static void main(String[] args) {
79
80          TrieTree trieTree = new TrieTree();
81
82          trieTree.insertString("hehao");
83          trieTree.insertString("ehaoh");
84          trieTree.insertString("haohe");
85          trieTree.insertString("aoheh");
86          trieTree.insertString("facri");
87          trieTree.insertString("et");
88          trieTree.insertString("oheha");
89
90          boolean result;
91
92          result = trieTree.searchString("hehao");
93          result = trieTree.searchString("et");
94          result = trieTree.searchString("asglajs");
95          result = trieTree.searchString("oaehh");
96
97          System.out.println("haha");
98      }
99  }
```