[LeetCode]422. Valid Word Square

Given a sequence of words, check whether it forms a valid word square.

A sequence of words forms a valid word square if the k^{th} row and column read the exact same string, where $0 \le k < \max(numRows, numColumns)$.

Note:

1. The number of words given is at least 1 and does not exceed 500.
2. Word length will be at least 1 and does not exceed 500.
3. Each word contains only lowercase English alphabet a-z.

```
Input:
[
  "abcd",
  "bnrt",
  "crmy",
  "dtye"
]

Output:
true

Explanation:
The first row and first column both read "abcd".
The second row and second column both read "bnrt".
The third row and third column both read "crmy".
The fourth row and fourth column both read "dtye".

Therefore, it is a valid word square.
```

Example 2:

```
Input:
[
  "abcd",
  "bnrt",
  "crm",
  "dt"
]

Output:
true

Explanation:
The first row and first column both read "abcd".
The second row and second column both read "bnrt".
The third row and third column both read "crm".
The fourth row and fourth column both read "dt".

Therefore, it is a valid word square.
```

```
Input:
[
  "ball",
  "area",
  "read",
  "lady"
]

Output:
false

Explanation:
The third row reads "read" while the third column reads "lead".

Therefore, it is NOT a valid word square.
```

题目大意：(吐槽一下，这题目真是又臭又长)这道题的意思是检测第i行中第j个字符是否等于第j行的第i个字符，从题目的意思其实我们用两个for循环遍历就能解决。
个人觉得这道题真正的考点是给你的第二个例子，就是各种边界条件你是否能考虑全面。我个人看下来需要考虑的有这么几个点：
然后在讨论区看到了一个非常好的思路，
就是说先把每一行转化为一个string，然后用行和列来进行对比，这样就不需要考虑那么多的边界条件。这样一个equals方法就能让我们得到结果。

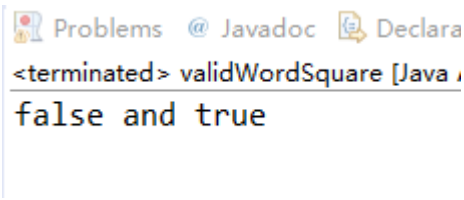
```java
import java.util.LinkedList;
import java.util.List;

public class validWordSquare {

public static boolean validWordSquareT(List<String> words) {
       for(int i=0;i<words.size();++i)
       {
       String s = words.get(i);
       String s2 = getVertical(i,words);
       if(!s.equals(s2)) return false;
       }
       return true;
    }

private static String getVertical(int col,List<String> words){
StringBuilder sb = new StringBuilder();
for(int i=0;i<words.size();++i)
{
     String word = words.get(i);
     if(col<word.length())
     {
          sb.append(word.charAt(col));
     }
}
     return sb.toString();
}
public static void main(String[] args) {
// TODO Auto-generated method stub
/*
  "ball",
  "area",
  "read",
  "lady"
 */
     List<String> wordslist1;
     List<String> wordslist2;
     String[] testStr = {"ball","area","read","lady"};
     String[] testStr2= {"abcd","bnrt","crm","dt"};
     wordslist1 = new LinkedList<String>();wordslist2 = new
     LinkedList<String>();
for(int k=0;k<testStr.length;++k) wordslist1.add(testStr[k]);
for(int k=0;k<testStr2.length;++k) wordslist2.add(testStr2[k]);
     boolean ans = validWordSquareT(wordslist1);
     boolean ans2 = validWordSquareT(wordslist2);
     System.out.printf("%b and %b\n",ans,ans2);
}
```

```
}
```


Problems
Javadoc
Declara
<terminated> validWordSquare [Java
false and true