JustAnotherStringProblem

This problem was asked by Dropbox.

Given a string `s` and a list of words `words`, where each word is the same length, find all starting indices of substrings in `s` that is a concatenation of every word in `words` exactly once.

For example, given `s = "dogcatcatcodecatcatdog"` and `String[] words1 = {"dog", "cat"}`, return `int[0, 16]`, since `"dogcat"` starts at index 0 and `"catdog"` starts at index 16.

Given `s = "barfoobazbitbyte"` and `String[] words2 = ["dog", "cat"]`, return `int[]` (an empty array) since there the strings `"dogcat"` and `"catdog"` are not substrings of `"barfoobazbitbyte"`.

In this problem you will complete the constructor and two methods in `JustAnotherStringProblem` class. The constructor has a single `String[]` parameter. You will implement two methods. The first method is the `getAllPermutations()` which returns a `List<String>` containing all permutation of the words passed to the `JustAnotherStringProblem` class in the constructor. The second method is previously described `getIndexes(String w)` method which returns the `int[]`.

The `getAllPermutations()` returns a `List<String>` containing all permutations (**in any order**) of the words passed to the `JustAnotherStringProblem` class in the constructor.

The following code shows the results of the `getAllPermutations` method.

The following code	Returns

`String[] words1 = {"dog", "cat"};`
`JustAnotherStringProblem jasp =`
`new JustAnotherStringProblem(words1);`
`int[] indexes = jasp.getAllPermutations();`	

`ans.size();`	`2`
`ans.contains("dogcat");`	`true`
`ans.contains("catdog");`	`true`

The following code shows additional results of the `getAllPermutations` method.

The following code	Returns

`String [] words2 = {"a", "b", "c"};`
`JustAnotherStringProblem jasp =`
`jasp = new JustAnotherStringProblem(words2);`
`List<String> ans =jasp.getAllPermutations();`	

`ans.size();`	`6`
`ans.contains("abc");`	`true`
`ans.contains("acb");`	`true`
`ans.contains("bac");`	`true`
`ans.contains("bca");`	`true`
`ans.contains("cab");`	`true`
`ans.contains("cba");`	`true`

The following code shows additional results of the `getAllPermutations` method.

The following code	Returns

`String[] words3 = {"a1", "b2", "c3", "d4"};`
 	
`jasp = new JustAnotherStringProblem(words3);`
 	
`ans =jasp.getAllPermutations();`	

`ans.size();`	`24`

The `getIndexes(String w)` returns a **sorted** `int[]` containing all starting indices of substrings in `s` that is a concatenation of every word in `words` exactly once.

See next page for sample runs!

The following code shows the results of the `getIndexes` method.

The following code	Returns

`String[] words1a = {"dog", "cat"};`
 	
`JustAnotherStringProblem jasp =`
`new JustAnotherStringProblem(words1a);`	

`int[] indexes = jasp.getIndexes("dogcatcatcodecatcatdog");`	
`indexes.length;`	2
`indexes[0];`	0
`indexes[1];`	16

`indexes = jasp.getIndexes("barfoobazbitbyte");`	
`indexes.length;`	0

`indexes = jasp.getIndexes("dogcatdogcatcodecatdog");`	
`indexes.length;`	4
`indexes[0];`	0
`indexes[1];`	3
`indexes[2];`	6
`indexes[3];`	16