1369. Most Common Word

Given a paragraph and a list of banned words, return the most frequent word that is not in the list of banned words. It is guaranteed there is at least one word that isn't banned, and that the answer is unique.

Words in the list of banned words are given in lowercase, and free of punctuation. Words in the paragraph are not case sensitive. The answer is in lowercase.

Example

Input:

paragraph = "Bob hit a ball, the hit BALL flew far after it was hit."

banned = ["hit"]

Output:"ball"

Explanation:

"hit" occurs 3 times, but it is a banned word.

"ball" occurs twice (and no other word does), so it is the most frequent non-banned word in the paragraph.

Note that words in the paragraph are not case sensitive,

that punctuation is ignored (even if adjacent to words, such as "ball,"),

and that "hit" isn't the answer even though it occurs more because it is banned.

Notice

* 1 <= paragraph.length <= 1000.
* 1 <= banned.length <= 100.
* 1 <= banned[i].length <= 10.
* The answer is unique, and written in lowercase (even if its occurrences in paragraph may have uppercase symbols, and even if it is a proper noun.)
* paragraph only consists of letters, spaces, or the punctuation symbols !?',;.
* Different words in paragraph are always separated by a space.
* There are no hyphens or hyphenated words.
* Words only consist of letters, never apostrophes or other punctuation symbols.

<https://www.lintcode.com/problem/most-common-word/description>

1. */\**
2. *\* To change this license header, choose License Headers in Project Properties.*
3. *\* To change this template file, choose Tools | Templates*
4. *\* and open the template in the editor.*
5. *\*/*
6. **package** javaapplication66;
8. **import** java.util.ArrayList;
9. **import** java.util.Collections;
10. **import** java.util.Comparator;
11. **import** java.util.HashMap;
12. **import** java.util.HashSet;
13. **import** java.util.LinkedHashMap;
14. **import** java.util.LinkedList;
15. **import** java.util.List;
16. **import** java.util.Map;
18. ***/\*\****
19. ***\****
20. ***\* @author Usuario***
21. ***\*/***
22. **public** **class** JavaApplication66 {

25. **public** **static** HashMap<String, Integer> sortByValue(HashMap<String, Integer> hm)
26. {
27. *// Create a list from elements of HashMap*
28. List<Map.Entry<String, Integer> > list =
29. **new** LinkedList<Map.Entry<String, Integer> >(hm.entrySet());
31. *// Sort the list*
32. Collections.sort(list, **new** Comparator<Map.Entry<String, Integer> >() {
33. **public** **int** compare(Map.Entry<String, Integer> o1,
34. Map.Entry<String, Integer> o2)
35. {
36. **return** (o2.getValue()).compareTo(o1.getValue());
37. }
38. });
40. *// put data from sorted list to hashmap*
41. HashMap<String, Integer> temp = **new** LinkedHashMap<String, Integer>();
42. **for** (Map.Entry<String, Integer> aa : list) {
43. temp.put(aa.getKey(), aa.getValue());
44. }
45. **return** temp;
46. }
48. **public** String mostCommonWord(String paragraph, String[] banned) {
49. *//*
50. String[] arr = paragraph.split(" ");
52. **for**(**int** i =0; i<arr.length; i++) {
53. arr[i] = arr[i].toLowerCase().trim();
54. **char** ultimo = arr[i].charAt(arr[i].length() - 1);
55. *//if( ultimo == ',' || ultimo == '.' || ultimo){*
56. **if**(!Character.isLetter(ultimo)){
57. arr[i] = arr[i].substring(0, arr[i].length() - 1);
58. }
59. }
61. HashMap<String, Integer> hash\_parag =
62. **new** HashMap();
63. **for**(**int** i =0; i<arr.length; i++) {
64. **if**(hash\_parag.containsKey(arr[i])) {
65. hash\_parag.put(arr[i], hash\_parag.get(arr[i]) + 1);
66. }
67. **else**
68. {
69. hash\_parag.put(arr[i], 1);
70. }
71. }
73. HashMap<String, Integer> ord\_hash\_parag = sortByValue( hash\_parag );
75. HashSet<String> hash\_banned =
76. **new** HashSet();
78. **for**(**int** i =0; i<banned.length; i++) {
79. hash\_banned.add(banned[i]);
80. }

83. **for**(String key: ord\_hash\_parag.keySet()) {
84. **if**(!hash\_banned.contains(key)) {
85. **return** key;
86. }
87. }
89. **return** "";
90. }

93. **public** **static** **void** main(String[] args) {
94. *// TODO code application logic here*
96. String paragraph = "Bob hit a ball, the hit BALL flew far after it was hit.";
97. String[] banned =  {"hit"};
99. String[] arr = paragraph.split(" ");
101. **for**(**int** i =0; i<arr.length; i++) {
102. arr[i] = arr[i].toLowerCase().trim();
103. **char** ultimo = arr[i].charAt(arr[i].length() - 1);
104. **if**( ultimo == ',' || ultimo == '.'){
105. arr[i] = arr[i].substring(0, arr[i].length() - 1);
106. }
107. }
109. HashMap<String, Integer> dic =
110. **new** HashMap();
112. **for**(**int** i =0; i<arr.length; i++) {
113. **if**(dic.containsKey(arr[i])) {
114. dic.put(arr[i], dic.get(arr[i]) + 1);
115. }
116. **else**{
117. dic.put(arr[i], 1);
118. }
119. }
121. HashMap<String, Integer> hash = sortByValue(dic);
123. **for**(String key: hash.keySet()) {
124. System.out.println(key + " " + hash.get(key) );
125. }
127. }
129. }