Vocab – AP Computer Science

Some applications use a controlled vocabulary to describe, or tag, things. A controlled vocabulary is a limited set of keywords from which appropriate tags can be chosen. The Vocab class, shown below, contains methods used to analyze words in terms of their presence in a controlled vocabulary. You will write three methods of the Vocab class.

part (a)

Write the helper method, *findWord*. The *findWord* method searches for a

string in array *theVocab*, returning true if a match is found and returning false otherwise.

part (b)

The *countNotInVocab* method returns an int that contains the number of words in its parameter *wordArray* that are **not** found in the instance variable *theVocab*.

part (c)

The *notInVocab* method returns an array of String objects that contains only elements of

its parameter *wordArray* that are **not** found in *theVocab*. Assume that there are no duplicates in *wordArray*.

The following example illustrates the behavior of the *notInVocab* method.

theVocab: {"time", "food", "dogs", "cats", "health", "plants","sports"}

wordArray: {"dogs", "toys", "sun", "plants", "time"}

Array returned by notInVocab: {"toys", "sun"}

/\*\*\* <--- complete 3 methods in the class below ---> \*/

import java.util.Arrays;

public class Vocab {

/\*\* The controlled vocabulary for a Vocab object. \*/

private String[] theVocab;

public Vocab(String[] words) {

theVocab = words;

}

/\*\* Searches for a string in theVocab. Returns true if its String

\* parameter str is an exact match to an element in theVocab and

\* returns false otherwise.

\*/

public boolean findWord(String str)

{ /\* to be implemented in part (a) \*/ }

/\*\* Counts how many strings in wordArray are not found in

\* theVocab, as described in part (b)

\*/

public int countNotInVocab(String[] wordArray)

{ /\* to be implemented in part (b) \*/ }

/\*\* Returns an array containing strings from wordArray not found in theVocab,

\* as described in part (c).

\*/

public String[] notInVocab(String[] wordArray)

{ /\* to be implemented in part (c) \*/

/\*\*\* The main tester method is complete. \*\*\*/

public static void main(String[] args)

{

String[] voc = {"time","food","dogs","cats","health","plants","sports"};

String[] words = {"cats","cookies","plants","computers","sports","pickles"};

Vocab v1 = new Vocab(voc);

System.out.println("theVocab: " + Arrays.toString(voc));

System.out.println("words: " + Arrays.toString(words));

System.out.println("\n---run---");

System.out.println("find \"cats\" in theVocab " + v1.findWord("cats"));

System.out.println("find \"pizza\" in theVocab " + v1.findWord("pizza"));

System.out.println("number of words not found in theVocab: " +

v1.countNotInVocab(words));

System.out.println("list of words not found in theVocab: " +

Arrays.toString(v1.notInVocab(words)));

System.out.println("\n---expected---");

System.out.println("find \"cats\" in theVocab true");

System.out.println("find \"pizza\"in theVocab false");

System.out.println("number of words not found in theVocab: 3");

System.out.println("list of words not found in theVocab: [cookies, computers, pickles]");

}

}