Homework 9 – Due May 30th 23:59, KST

Instructions: Complete the implementation and turn it in before the due date. Any deviations from the instructed deliverable format will result in a deduction of grade. DO NOT COPY OTHER'S WORKS!

You will be implementing a very simple spell checker for this assignment. In particular, you are to implement a method that returns all possible corrections to one or more typos. A typo is characterized by the absence from the dictionary that will be provided to you. The input will be a sentence of English words, and the output will be a set of candidate corrections for all typos.
The basic algorithm is as follows: Store all valid words in the dictionary. For each word in the given sentence, see if the word is valid. If it is valid, then you don't have to provide any candidates. If it is invalid (i.e., a typo) you should prepare a list of valid correction candidates and make it part of the final output. The valid correction candidates are prepared by the following modifications.

- Removal of a single character from any position: "edog" → "dog", "eog", "edg", "edo"
- Addition of a single character into any position: "aun" → "aaun", "aunt", "jaun", "auzn"…
- Replacement of a single character: "abple" → "apple", "bbple", "abpke", …
- Swapping two adjacent characters: "uant" → "aunt", "unat", "uatn"

For each of the candidate correction, you should test whether it's part of the dictionary to determine if it indeed is a valid candidate. The time complexity of checking whether the given word is valid MUST be as efficient as possible. You must also implement any data structures you need on your own.

Implementation: Implement the method spellCheck(String) that receives a sentence in String. Return an array of ArrayList<String> that contains correction candidates for each word. More precisely, each position of the array should be an ArrayList<String> that lists all candidates for the word corresponding to that position in the input sentence. If the word is valid, then it should be null. For example, if the input sentence is "I love my uant", the output should be {null, null, null, [aunt, ant]}, where [] indicates ArrayList's contents. This is assuming that 'aunt' and 'ant' are part of the dictionary and 'uant' is not. The exact output of this example will depend on the dictionary I give you. It doesn't matter if a word is correct or not in real world: only the words in the dictionary file are valid ones.

Rubric: Grading will be based on, but not limited to, the following criteria.

- Documentation (40 points): For the required methods, you should provide extensive descriptions in the header comments on your approaches. In particular, analyze the time complexity in detail.

- Correctness (60 points): Your implementation should correctly retrieve the candidate corrections. 40 of the 60 points will be given to the correct outputs of the test cases. The remaining 20 points will be given to those who come up with the best time complexity.
- Miscellaneous: Do not change the method and class names or declare a new package. Do not use packages other than the ones provided at the top of the file. You must submit an ***error-free*** program that will compile without any syntax errors. Two or more unexpected exceptions will result in a zero (0) for correctness. As always, do NOT consult anyone or any references outside of this class.

Deliverable: A single SpellCheck.java file not part of any package structures. Do NOT rename the file.