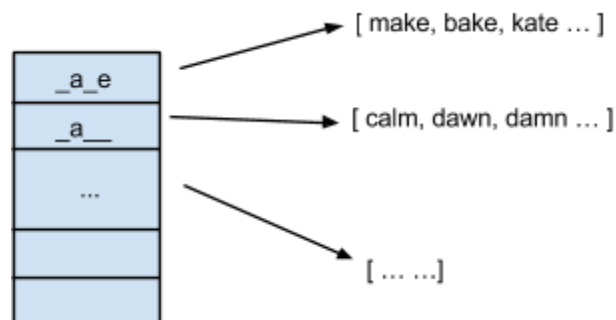


If you are creating new test cases, make sure all the words are covered by “corpus”. If not, please manually add them into “corpus”. “fruitninja” and “aspca” are not automatically covered.

Algorithm:

The first part of the algorithm is preprocessing the dictionary and grouping of words with the same pattern. A pattern represents relative positions of vowels in a word. For example, The pattern for both “make” and “bake” is “_a_e”. Therefore, words with the same patterns can be grouped together and stored in the same hash slot of a hash table. Here is a graphical demonstration:



Before processing the dictionary, scan the board and find all the patterns. Then find all the words that match those patterns. Assume the dictionary is of size $O(N)$, then by doing this the search space for each word is shrunk to $O(\frac{N}{\alpha})$, while $\alpha > 1$.

The second part is to recursively try all different combinations as well as backtracking. The algorithm follows a DFS pattern to move from one incomplete word to another. Here is a graphical demonstration:



Assume that a board contains M incomplete words, then the backtracking algorithm yields $O((\frac{N}{\alpha})^M)$ runtime complexity as well as $O(N)$ space complexity for storing the hashmap.

Most of crossword puzzles are solved within one minute using this program.