Problem #211 : Design Add and Search Words Data Structure

<https://leetcode.com/problems/design-add-and-search-words-data-structure/>

Discussion:

<https://leetcode.com/problems/design-add-and-search-words-data-structure/discuss/776548/Simple-Python-3-Solution-without-Trie-beats-96.56>

Simple Python 3 Solution without Trie -- beats 96.56%

1. I found that regular expression does not seem to match '.' with a single character that is not a newline. I had to use length of the word as well.
2. I tried using a set but got "Time Limit Exceeded" with this answer.

class WordDictionary:

def \_\_init\_\_(self):

"""

Initialize your data structure here.

"""

self.wordDict = set()

def addWord(self, word: str) -> None:

"""

Adds a word into the data structure.

"""

self.wordDict.add(word)

def search(self, word: str) -> bool:

"""

Returns if the word is in the data structure. A word could contain the dot character '.' to represent any one letter.

"""

if '.' in word:

n = len(word)

return(any(len(x) == n and re.findall(word, x) for x in self.wordDict))

else:

return(word in self.wordDict)

---------------------------------------

1. Then, I used a dictionary data structure with length of the word as key and the words in a list as value. This worked! This beats 96.56% in runtime.

class WordDictionary:

def \_\_init\_\_(self):

"""

Initialize your data structure here.

"""

self.wordDict = {}

def addWord(self, word: str) -> None:

"""

Adds a word into the data structure.

"""

if len(word) not in self.wordDict.keys():

self.wordDict[len(word)] = [word]

else:

self.wordDict[len(word)].append(word)

def search(self, word: str) -> bool:

"""

Returns if the word is in the data structure. A word could contain the dot character '.' to represent any one letter.

"""

n = len(word)

if n not in self.wordDict.keys():

return(False)

if '.' not in word:

return(word in self.wordDict[n])

else:

rex = re.compile(word)

return(any(rex.findall(x) for x in self.wordDict[n]))

# Your WordDictionary object will be instantiated and called as such:

# obj = WordDictionary()

# obj.addWord(word)

# param\_2 = obj.search(word)

----------------------------------------------