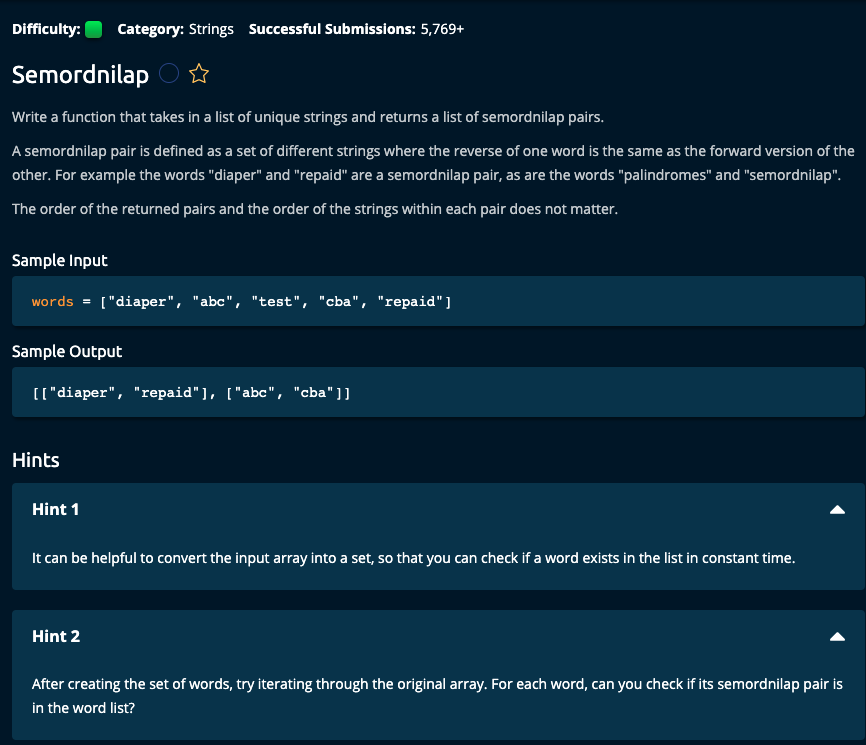
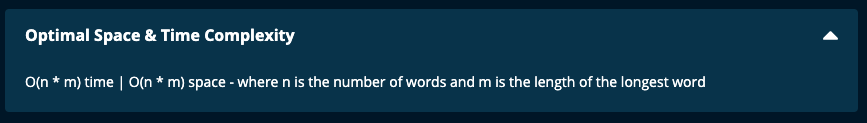
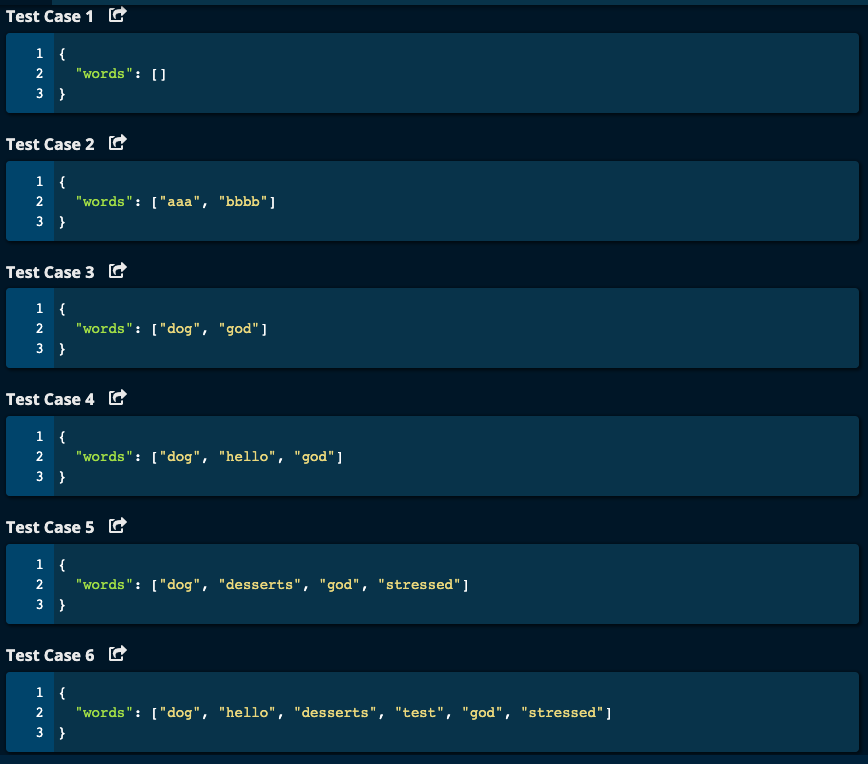
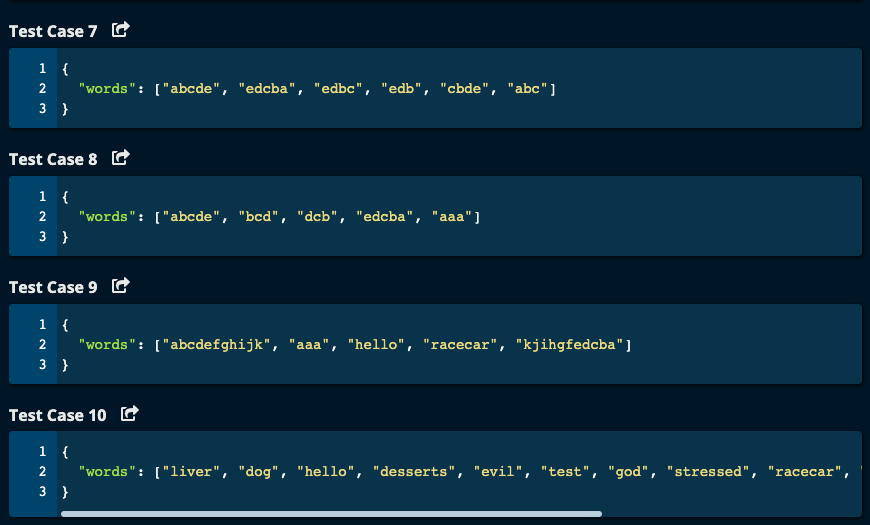
Semordnilap (Easy)









My Solution: O(n\*m) time | O(n \*m) space where n = length of words array and m = length of the longest word in words array

def semordnilap(words):

if len(words) == 0:

return []

result = []

wordDict = {}

for word in words:

if word not in wordDict:

wordDict[word[::-1]] = word

else:

result.append([word, wordDict[word]])

return result

JJ Notes:

1. If length of the array words is 0, then return an empty list.
2. Initialize result to an empty list.
3. Initialize wordDict to an empty dictionary.
4. Iterate through the words array. For each word in the words array, if it is not already in wordDict, then reverse the word and make it the key in wordDict and the value for this key is the word itself. If the word is already in the dictionary, then append to the result a list consisting of the word and its reverse (obtained from the dictionary).
5. Return result array.

Algoexpert Solution: O(n\*m) time | O(n \*m) space where n = length of words array and m = length of the longest word in words array

# Algoexpert Solution - O(n \* m) Time | O(n \* m) Space where n = length of words array

# and m = length of the longest word

def semordnilap(words):

wordSet = set(words)

semordnilapPairs = []

for word in words:

reverse = word[::-1]

if reverse in wordSet and reverse != word: # we don't want a palindrome

semordnilapPairs.append([word, reverse])

wordSet.remove(word)

wordSet.remove(reverse)

return semordnilapPairs

JJ Notes:

1. Create a wordSet with words array.
2. Initialize semordnilapPairs as an empty list to hold the results.
3. Iterate through words array. For each word in words array, get the reverse of the word.

If the reverse is in words array and word is not a palindrome, then create a list with the word and reverse and append this list to semordnilapPairs list. Then remove word and reverse from wordSet (since we want to avoid redundancy in semordnilapPairs).

1. Return semordnilapPairs.