**Problem #2062: Count Vowel Substrings of a String (Easy)**

<https://leetcode.com/problems/count-vowel-substrings-of-a-string/description/>

**My Solution:**

1. Let vowel\_set be set of “aeiou”.
2. Initialize count to 9.
3. For I in range of length of word, initialize alist to an empty list.

If word at index I is a vowel, then append it to alist.

Now iterate j from index I + 1 to length of word. If word at index j is a vowel, then append it to alist. Check of alist has all the vowels in vowel\_set and if so, increment count. If word at index j is not vowel, break from the inner loop witj j.

1. At the end, return count.

class Solution:

def countVowelSubstrings(self, word: str) -> int:

vowel\_set = set("aeiou")

count = 0

for i in range(len(word)):

alist = []

if word[i] in vowel\_set:

alist.append(word[i])

for j in range(i + 1, len(word)):

if word[j] in vowel\_set:

alist.append(word[j])

if set(alist) == vowel\_set:

count += 1

else:

break

#print("i = ", i, "j = ", j, "word[i] = ", word[i], "word[j] = ", word[j])

#print("alist = ", alist, "count = ", count)

return count