

## C'mon K-mers!

A useful operation on strings, specially when analyzing certain types of data, like genomic data, is to find all the substrings of length  $k$  in a string. These substrings are called the string's  $k$ -mers and we are typically interested in finding out how many times each  $k$ -mer appears within the string. For example, consider the following string:

aaabaab

This string has three unique 2-mers:

aaabaab

aa  
  aa  
    ab  
      ba  
        aa  
          ab

More specifically, the 2-mers are **aa** (appearing three times), **ab** (appearing two times), and **ba** (appearing once). Notice how counting the  $k$ -mers considers *overlapping* occurrences of each substring. For example, the first two occurrences of **aa** above overlap with each other.

We could do a similar analysis for the string's 3-mers:

aaabaab

aaa  
  aab  
    aba  
      baa  
        aab

In this case, there are four unique 3-mers: **aaa** (appearing once), **aab** (appearing twice), **aba** (appearing once), and **baa** (appearing once).

In this problem, you will write a function `kmers` that, given a potentially large string, a value for  $k$ , and some  $k$ -mers, will determine how many times each  $k$ -mer appears in the string. The function should return a list of integers where the  $i$ th integer is the number of times the  $i$ th mer in the list of mers occurs in the sequence.

Here are some sample inputs and outputs:

Sequence	K-Mers	Results
aaabaab	ab ba aa	2 1 3

Sequence	K-Mers	Results
aaabaab	aaa aab aaa baa xyz	1 2 1 1 0

Here is the skeleton code for this task:

```
'''
Distribution file of the kmers problem.
'''

from typing import List

def kmers(seq: str, k: int, k_mers: List[str]) -> List[int]:
    '''
    Takes in a sequence of letters, a kmer length, and a list of k-mers and
    returns a list of the counts of each k-mer in the sequence.

    Parameters:
        - seq (str): A string of letters
        - k (int): The length of the k-mers
        - k_mers (list): A list of k-mers

    Returns:
        - k_mer_counts (list): A list of the counts of each k-mer in the
          sequence. The counts should be in the same order as the k-mers in
          the k_mers list.
    '''
    # TODO: Implement this function.

    return []
```

When you take the placement exam, you will be expected to copy the skeleton code into a file and then complete the function. For the practice problems, we have provided a file named problem4.py that includes the above header for your convenience.