



Daily Coding Problem #57

Problem

This problem was asked by Amazon.

Given a string *s* and an integer *k*, break up the string into multiple texts such that each text has a length of *k* or less. You must break it up so that words don't break across lines. If there's no way to break the text up, then return null.

You can assume that there are no spaces at the ends of the string and that there is exactly one space between each word.

For example, given the string "the quick brown fox jumps over the lazy dog" and *k* = 10, you should return: ["the quick", "brown fox", "jumps over", "the lazy", "dog"]. No string in the list has a length of more than 10.

Solution

We can break up the string greedily. First we'll break up *s* into an array *words*. Then, we can use a buffer as a current string and tentatively add words to it, checking that the newly-added-to line can fit within *k*. If we overflow with the new word, then we flush out the current string into an array *all* and restart it with the new word.

Notice that if any word is longer than *k*, then there's no way to break up the text, so we should return None. It's helpful to define a helper function that returns the length of a list of words with spaces added in between.

Finally, we return *all*, which should contain the texts we want.

```
def break(s, k):  
    words = s.split()  
  
    if not words:  
  
    return []
```

```
current = []
all = []

for i, word in enumerate(words):
    if length(current + [word]) <= k:
        current.append(word)
    elif length([word]) > k:
        return None
    else:
        all.append(current)
        current = [word]
all.append(current)

return all

def length(words):
    if not words:
        return 0
    return sum(len(word) for word in words) + (len(words) - 1)
```

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)
