Repeated DNA Sequences

Try to solve the Repeated DNA Sequences problem.

We'll cover the following

Statement

Given a string, `s`, that represents a DNA subsequence, and a number k, return all the contiguous subsequences (substrings) of length k that occur more than once in the string. The order of the returned subsequences does not matter. If no repeated substring is found, the function should return an empty set.

> The DNA sequence is composed of a series of nucleotides abbreviated as A, C, G, and T. For example, $ACGAATTCCG$ is a DNA sequence. When studying DNA, it is useful to identify repeated sequences in it.

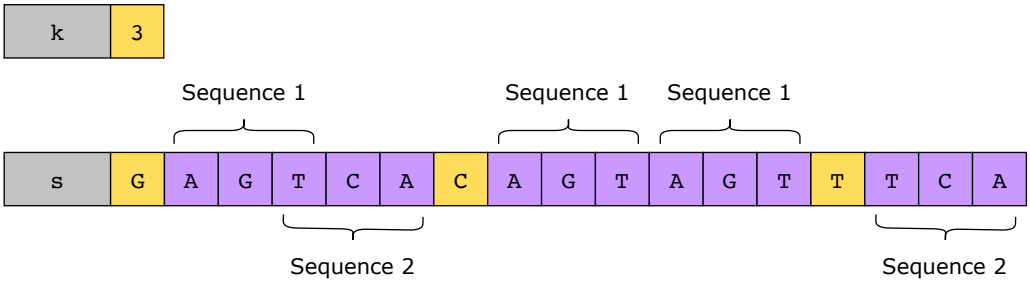
Constraints:

- $1 \leq$ `s.length` $\leq 10^4$
- `s[i]` is either A, C, G, or T.

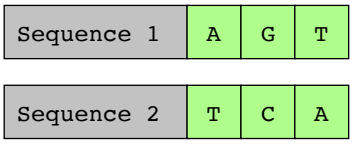
Examples

Sample example 1

Input



Output



Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Repeated DNA Sequences

1 **(Select all that apply.)** What is the output if the following string and value of k are given as input?

s = "AGCTGAAAGCTTAGCTG"

k = 5

- [ ] **A)** AGCTG
- [ ] **B)** TTAGC
- [ ] **C)** AAGCT
- [ ] **D)** AAAGC

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

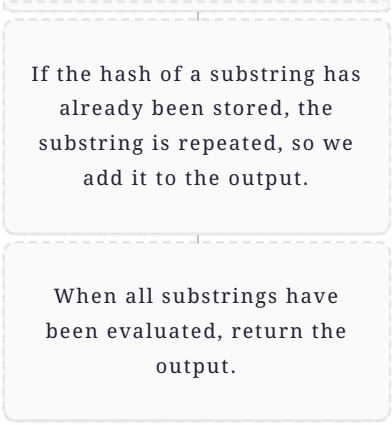
Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Iterate over the k-length substrings of the input string.

Store the hash of the current substring to keep track of all unique substrings.



Reset | Show Solution | Submit

Try it yourself

Implement your solution in the following coding playground:

Java

```
2
3  public class FindRepeatedSequences{
4
5      public static Set<String> findRepeatedSequences(String s, int k) {
6
7       // Your code will replace this placeholder return statement
8       return new HashSet<>();
9      }
10 }
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

"AAAAACCCCCAAAAACCCCCC"

Input #2

8

Repeated DNA Sequences