Find All Anagrams in a String

Try to solve the Find All Anagrams in a String problem.

We'll cover the following

Statement

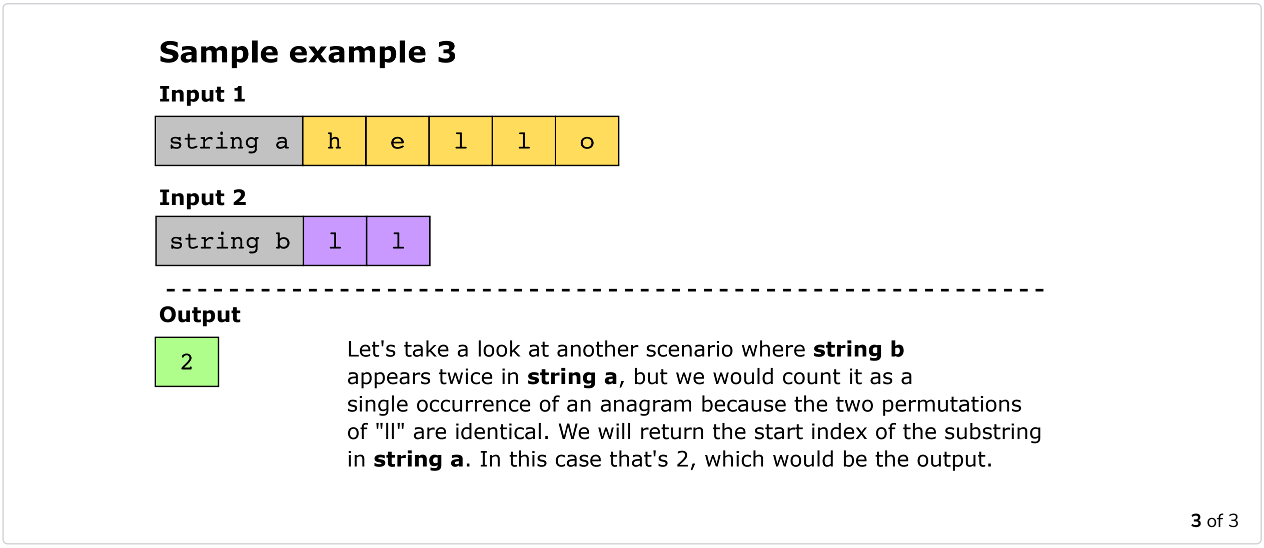
Given two strings, `a` and `b`, return an array of all the start indexes of anagrams of `b` in `a`. We may return the answer in any order.

> An **anagram** is a word or phrase created by rearranging the letters of another word or phrase while utilizing each of the original letters exactly once. For example, “act” is the anagram of “cat”, and “flow” is the anagram of “wolf”.

Constraints:

- $1 \leq$ `a.length`, `b.length` $\leq 3 \times 10^3$
- Both `a` and `b` consist only of lowercase English letters.

Examples



Sample example 3

Input 1

string a	h	e	l	l	o

Input 2

string b	l	l

Output

2

Let's take a look at another scenario where **string b** appears twice in **string a**, but we would count it as a single occurrence of an anagram because the two permutations of "ll" are identical. We will return the start index of the substring in **string a**. In this case that's 2, which would be the output.

3 of 3

Understand the problem

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

Find All Anagrams in a String

1

What’s the correct output for the following inputs?

`a` = “abab”

`b` = “ab”

A) [0]

B) [0, 2]

C) [0, 1, 2]

D) [1, 2]

Submit Answer

Question 1 of 5
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.

Create two hash maps, `hashA` and `hashB`, acting as counters. `hashB` stores the frequency of characters in string `b`, and `hashA` stores the count of characters in the sliding window inside the string `a`.

While traversing string `a`, if (at any point) `hashA` equals `hashB`, it means we found an anagram. Therefore, add the index of the first character of the current sliding window to the output array.

After traversing the entire string a, return the array of all the start indexes of the anagrams of string b in string a as the output.

Otherwise, if we don't find any anagrams of string b in string a, return an empty array.

Reset Show Solution Submit

Try it yourself

Implement your solution in the following coding playground:

usercode > main.java

```java
import java.util.*;
public class Main{
    public static List<Integer> findAnagrams(String a, String b) {

        // Replace this placeholder return statement with your code
        return new ArrayList<>();
    }
}
```

Powered by AI Submit

Test Cases Results

Case 1 Case 2 Case 3

Input #1

"abab"

Input #2

"ab"

Find All Anagrams in a String

Back
Solution: First Unique ...

Next
Solution: Find All Ana...

✓ Mark as Completed