Problem 1455: Check If a Word Occurs As a Prefix of Any Word in a Sentence

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Given a

sentence

that consists of some words separated by a

single space

, and a

searchWord

, check if

searchWord

is a prefix of any word in

sentence

.

Return

the index of the word in

sentence

(

1-indexed

) where

searchWord

is a prefix of this word

. If

searchWord

is a prefix of more than one word, return the index of the first word

(minimum index)

. If there is no such word return

-1

.

A

prefix

of a string

s

is any leading contiguous substring of

s

.

Example 1:

Input:

sentence = "i love eating burger", searchWord = "burg"

Output:

4

Explanation:

"burg" is prefix of "burger" which is the 4th word in the sentence.

Example 2:

Input:

sentence = "this problem is an easy problem", searchWord = "pro"

Output:

2

Explanation:

"pro" is prefix of "problem" which is the 2nd and the 6th word in the sentence, but we return 2 as it's the minimal index.

Example 3:

Input:

sentence = "i am tired", searchWord = "you"

Output:

-1

Explanation:

"you" is not a prefix of any word in the sentence.

Constraints:

1 <= sentence.length <= 100

1 <= searchWord.length <= 10

sentence

consists of lowercase English letters and spaces.

searchWord

consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int isPrefixOfWord(string sentence, string searchWord) {

}
};
```

Java:

```
class Solution {
public int isPrefixOfWord(String sentence, String searchWord) {

}
}
```

Python3:

```
class Solution:
def isPrefixOfWord(self, sentence: str, searchWord: str) -> int:
```

Python:

```
class Solution(object):
def isPrefixOfWord(self, sentence, searchWord):
"""
:type sentence: str
:type searchWord: str
:rtype: int
"""
```

JavaScript:

```
/**
* @param {string} sentence
* @param {string} searchWord
* @return {number}
*/
var isPrefixOfWord = function(sentence, searchWord) {

};
```

TypeScript:

```
function isPrefixOfWord(sentence: string, searchWord: string): number {

};
```

C#:

```
public class Solution {
public int IsPrefixOfWord(string sentence, string searchWord) {

}
}
```

C:

```
int isPrefixOfWord(char* sentence, char* searchWord) {

}
```

Go:

```
func isPrefixOfWord(sentence string, searchWord string) int {

}
```

Kotlin:

```
class Solution {
fun isPrefixOfWord(sentence: String, searchWord: String): Int {

}
}
```

Swift:

```
class Solution {
func isPrefixOfWord(_ sentence: String, _ searchWord: String) -> Int {

}
}
```

Rust:

```
impl Solution {
pub fn is_prefix_of_word(sentence: String, search_word: String) -> i32 {

}
}
```

Ruby:

```
# @param {String} sentence
# @param {String} search_word
# @return {Integer}
def is_prefix_of_word(sentence, search_word)

end
```

PHP:

```
class Solution {

/**
* @param String $sentence
* @param String $searchWord
* @return Integer
*/
function isPrefixOfWord($sentence, $searchWord) {

}
}
```

Dart:

```
class Solution {
int isPrefixOfWord(String sentence, String searchWord) {

}
}
```

Scala:

```
object Solution {
def isPrefixOfWord(sentence: String, searchWord: String): Int = {

}
}
```

Elixir:

```
defmodule Solution do
@spec is_prefix_of_word(sentence :: String.t, search_word :: String.t) ::
integer
def is_prefix_of_word(sentence, search_word) do

end
end
```

Erlang:

```
-spec is_prefix_of_word(Sentence :: unicode:unicode_binary(), SearchWord ::
unicode:unicode_binary()) -> integer().
is_prefix_of_word(Sentence, SearchWord) ->
.
```

Racket:

```
(define/contract (is-prefix-of-word sentence searchWord)
(-> string? string? exact-integer?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public:
int isPrefixOfWord(string sentence, string searchWord) {

}
};
```

Java Solution:

```
/**
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

class Solution {
public int isPrefixOfWord(String sentence, String searchWord) {

}
}
```

Python3 Solution:

```
"""
Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
Difficulty: Easy
Tags: array, string, tree, search

Approach: Use two pointers or sliding window technique
Time Complexity: O(n) or O(n log n)
Space Complexity: O(h) for recursion stack where h is height
"""

class Solution:
def isPrefixOfWord(self, sentence: str, searchWord: str) -> int:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def isPrefixOfWord(self, sentence, searchWord):
"""
:type sentence: str
:type searchWord: str
:rtype: int
"""
```

JavaScript Solution:

```
/**
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

/**
* @param {string} sentence
* @param {string} searchWord
* @return {number}
*/
var isPrefixOfWord = function(sentence, searchWord) {

};
```

TypeScript Solution:

```
/**
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

function isPrefixOfWord(sentence: string, searchWord: string): number {

};
```

C# Solution:

```
/*
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

public class Solution {
public int IsPrefixOfWord(string sentence, string searchWord) {

}
}
```

C Solution:

```
/*
* Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
* Difficulty: Easy
* Tags: array, string, tree, search
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(h) for recursion stack where h is height
*/

int isPrefixOfWord(char* sentence, char* searchWord) {

}
```

Go Solution:

```
// Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
// Difficulty: Easy
// Tags: array, string, tree, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

func isPrefixOfWord(sentence string, searchWord string) int {

}
```

Kotlin Solution:

```
class Solution {
fun isPrefixOfWord(sentence: String, searchWord: String): Int {

}
}
```

Swift Solution:

```
class Solution {
func isPrefixOfWord(_ sentence: String, _ searchWord: String) -> Int {

}
}
```

Rust Solution:

```
// Problem: Check If a Word Occurs As a Prefix of Any Word in a Sentence
// Difficulty: Easy
// Tags: array, string, tree, search
//
// Approach: Use two pointers or sliding window technique
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(h) for recursion stack where h is height

impl Solution {
pub fn is_prefix_of_word(sentence: String, search_word: String) -> i32 {

}
}
```

Ruby Solution:

```
# @param {String} sentence
# @param {String} search_word
# @return {Integer}
def is_prefix_of_word(sentence, search_word)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $sentence
* @param String $searchWord
* @return Integer
*/
function isPrefixOfWord($sentence, $searchWord) {

}
}
```

Dart Solution:

```
class Solution {
int isPrefixOfWord(String sentence, String searchWord) {

}
}
```

Scala Solution:

```
object Solution {
def isPrefixOfWord(sentence: String, searchWord: String): Int = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec is_prefix_of_word(sentence :: String.t, search_word :: String.t) ::
integer
def is_prefix_of_word(sentence, search_word) do

end
end
```

Erlang Solution:

```
-spec is_prefix_of_word(Sentence :: unicode:unicode_binary(), SearchWord ::
unicode:unicode_binary()) -> integer().
```

```
is_prefix_of_word(Sentence, SearchWord) ->
.
```

Racket Solution:

```
(define/contract (is-prefix-of-word sentence searchWord)
(-> string? string? exact-integer?)
)
```