

Problem 1910: Remove All Occurrences of a Substring

Problem Information

Difficulty: Medium

Acceptance Rate: 78.33%

Paid Only: No

Tags: String, Stack, Simulation

Problem Description

Given two strings `s` and `part`, perform the following operation on `s` until **all** occurrences of the substring `part` are removed:

* Find the **leftmost** occurrence of the substring `part` and **remove** it from `s`.

Return `s` after removing all occurrences of `part`.

A **substring** is a contiguous sequence of characters in a string.

Example 1:

Input: `s = "daabcbaabcbc", part = "abc"` **Output:** `"dab"` **Explanation:** The following operations are done: - `s = "da__abc__baabcbc"`, remove "abc" starting at index 2, so `s = "dabaabcbc"`. - `s = "daba__abc__bc"`, remove "abc" starting at index 4, so `s = "dababc"`. - `s = "dab__abc__"`, remove "abc" starting at index 3, so `s = "dab"`. Now `s` has no occurrences of "abc".

Example 2:

Input: `s = "axxxxyyyb", part = "xy"` **Output:** `"ab"` **Explanation:** The following operations are done: - `s = "axxx__xy__yyb"`, remove "xy" starting at index 4 so `s = "axxyyyb"`. - `s = "axx__xy__yyb"`, remove "xy" starting at index 3 so `s = "axxyyb"`. - `s = "ax__xy__yb"`, remove "xy" starting at index 2 so `s = "axyb"`. - `s = "a__xy__b"`, remove "xy" starting at index 1 so `s = "ab"`. Now `s` has no occurrences of "xy".

****Constraints:****

*`1` <= s.length <= 1000` *`1` <= part.length <= 1000` *`s`**■■■■■■■■** and `part` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {  
public:  
    string removeOccurrences(string s, string part) {  
  
    }  
};
```

Java:

```
class Solution {  
    public String removeOccurrences(String s, String part) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def removeOccurrences(self, s: str, part: str) -> str:
```