Minimum Window Subsequence

Try to solve the Minimum Window Subsequence problem.

We'll cover the following

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

Given two strings, `str1` and `str2`, find the shortest substring in `str1` such that `str2` is a subsequence of that substring.

A **substring** is defined as a contiguous sequence of characters within a string. A **subsequence** is a sequence that can be derived from another sequence by deleting zero or more elements without changing the order of the remaining elements.

Let's say you have the following two strings:

`str1` = "$abbcb$"

`str2` = "ac"

In this example, "$abbc$" is a substring of `str1`, from which we can derive `str2` simply by deleting both the instances of the character b. Therefore, `str2` is a subsequence of this substring. Since this substring is the shortest among all the substrings in which `str2` is present as a subsequence, the function should return this substring, that is, "$abbc$".

> If there is no substring in `str1` that covers all characters in `str2`, return an empty string.

> If there are multiple minimum-length substrings that meet the subsequence requirement, return the one with the left-most starting index.

Constraints:

- $1 \leq$ `str1.length` $\leq 2 \times 10^3$
- $1 \leq$ `str2.length` ≤ 100
- `str1` and `str2` consist of uppercase and lowercase English letters.

Examples

Sample example 1

Input

str1	"abcdebdde"
str2	"bde"

Output

Output String	"bcde"

The strings "bcde" and "bdde" are both minimum subsequences, but "bcde" occurs before "bdde".

The substring “deb” is the shortest to contain all the required characters, but they do not appear in the required order.

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:

Minimum Window Subsequence

1 What are valid substrings of “Educative”?

- [ ] **A)** “tive”
- [ ] **B)** “ude”
- [ ] **C)** “cat”
- [ ] **D)** “vit”

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.

Iterate over `str1` until all the characters of `str2` have been found in the same order and mark the end of the current window.

To find the smallest subsequence, iterate backward over `str1` until all the characters of `str2` have been found in the reverse order, and mark the start of the current window.

Update the minimum window subsequence if the current window is smaller than the shortest subsequence found so far.

Repeat the process, starting every time from the second character of the current window, until the end of `str1` has been reached.

Finally, return the minimum window subsequence.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in `main.java` in the following coding playground:

Java

```
2  public class Main{
3      public static String minWindow(String s, String t) {
4
5          // your code will replace the following placeholder return statement
6          return "";
7      }
8  }
```

Powered by AI

Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

"abcdebdde"

Input #2

"bde"

Minimum Window Subsequence

Back

Solution: Find Maximu...

Next

Solution: Minimum Wi...

Mark as Completed