Problem 727: Minimum Window Subsequence

Problem Information

Difficulty: Hard
Acceptance Rate: 43.69%
Paid Only: Yes
Tags: String, Dynamic Programming, Sliding Window

Problem Description

Given strings `s1` and `s2`, return _the minimum contiguous substring part of _`s1` _, so that_`s2` _is a subsequence of the part_.

If there is no such window in `s1` that covers all characters in `s2`, return the empty string `""`. If there are multiple such minimum-length windows, return the one with the **left-most starting index**.

Example 1:

Input: s1 = "abcdebdde", s2 = "bde" **Output:** "bcde" **Explanation:** "bcde" is the answer because it occurs before "bdde" which has the same length. "deb" is not a smaller window because the elements of s2 in the window must occur in order.

Example 2:

Input: s1 = "jmeqksfrsdcmsiwvaovztaqenprpvnbstl", s2 = "u" **Output:** ""

Constraints:

* `1 <= s1.length <= 2 * 104` * `1 <= s2.length <= 100` * `s1` and `s2` consist of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
string minWindow(string s1, string s2) {

}
};
```

Java:

```
class Solution {
public String minWindow(String s1, String s2) {

}
}
```

Python3:

```
class Solution:
def minWindow(self, s1: str, s2: str) -> str:
```