Problem 76: Minimum Window Substring

Problem Information

Difficulty: Hard
Acceptance Rate: 46.39%
Paid Only: No
Tags: Hash Table, String, Sliding Window

Problem Description

Given two strings `s` and `t` of lengths `m` and `n` respectively, return _the**minimum window**_ **_substring_** _of_`s` _such that every character in_`t` _(**including duplicates**) is included in the window_. If there is no such substring, return _the empty string_`""`.

The testcases will be generated such that the answer is **unique**.

Example 1:

Input: s = "ADOBECODEBANC", t = "ABC" **Output:** "BANC" **Explanation:** The minimum window substring "BANC" includes 'A', 'B', and 'C' from string t.

Example 2:

Input: s = "a", t = "a" **Output:** "a" **Explanation:** The entire string s is the minimum window.

Example 3:

Input: s = "a", t = "aa" **Output:** "" **Explanation:** Both 'a's from t must be included in the window. Since the largest window of s only has one 'a', return empty string.

Constraints:

* `m == s.length` * `n == t.length` * `1 <= m, n <= 105` * `s` and `t` consist of uppercase and lowercase English letters.

Follow up: Could you find an algorithm that runs in `O(m + n)` time?

Code Snippets

C++:

```
class Solution {
public:
string minWindow(string s, string t) {

}
};
```

Java:

```
class Solution {
public String minWindow(String s, String t) {

}
}
```

Python3:

```
class Solution:
def minWindow(self, s: str, t: str) -> str:
```