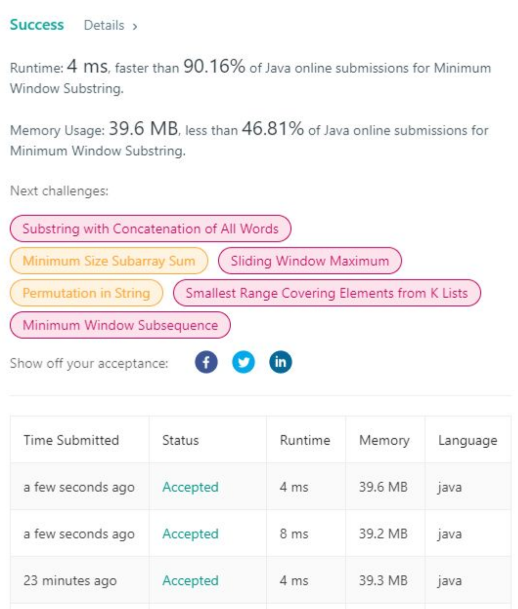
Student: Ovidiu Mura

Problem 76. Minimum Window Substring



Success Details ›

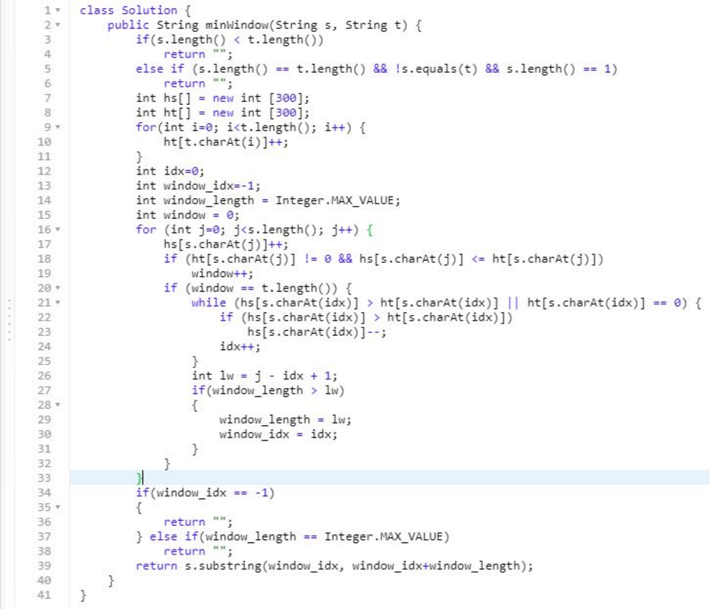
Runtime: 4 ms, faster than 90.16% of Java online submissions for Minimum Window Substring.

Memory Usage: 39.6 MB, less than 46.81% of Java online submissions for Minimum Window Substring.

Next challenges:

Substring with Concatenation of All Words
Minimum Size Subarray Sum
Sliding Window Maximum
Permutation in String
Smallest Range Covering Elements from K Lists
Minimum Window Subsequence
Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	4 ms	39.6 MB	java
a few seconds ago	Accepted	8 ms	39.2 MB	java
23 minutes ago	Accepted	4 ms	39.3 MB	java



```java
class Solution {
    public String minWindow(String s, String t) {
        if(s.length() < t.length())
            return "";
        else if (s.length() == t.length() && !s.equals(t) && s.length() == 1)
            return "";
        int hs[] = new int [300];
        int ht[] = new int [300];
        for(int i=0; i<t.length(); i++) {
            ht[t.charAt(i)]++;
        }
        int idx=0;
        int window_idx=-1;
        int window_length = Integer.MAX_VALUE;
        int window = 0;
        for (int j=0; j<s.length(); j++) {
            hs[s.charAt(j)]++;
            if (ht[s.charAt(j)] != 0 && hs[s.charAt(j)] <= ht[s.charAt(j)])
                window++;
            if (window == t.length()) {
                while (hs[s.charAt(idx)] > ht[s.charAt(idx)] || ht[s.charAt(idx)] == 0) {
                    if (hs[s.charAt(idx)] > ht[s.charAt(idx)])
                        hs[s.charAt(idx)]--;
                    idx++;
                }
                int lw = j - idx + 1;
                if(window_length > lw)
                {
                    window_length = lw;
                    window_idx = idx;
                }
            }
        }
        if(window_idx == -1)
        {
            return "";
        } else if(window_length == Integer.MAX_VALUE)
            return "";
        return s.substring(window_idx, window_idx+window_length);
    }
}
```