Minimum Window Substring

My Submissions (/problems/minimum-window-substring/submissions/)

Question Solution

Total Accepted: **43120** Total Submissions: **223651** Difficulty: **Hard**

> Given a string S and a string T, find the minimum window in S which will contain all the characters in T in complexity O(n).
>
> For example,
>
> **S** = `"ADOBECODEBANC"`
>
> **T** = `"ABC"`
>
> Minimum window is `"BANC"` .
>
> **Note:**
>
> If there is no such window in S that covers all characters in T, return the empty string `""` .
>
> If there are multiple such windows, you are guaranteed that there will always be only one unique minimum window in S.
>
> Show Tags
>
> Show Similar Problems

Have you met this question in a real interview? Yes No

Discuss (/discuss/questions/oj/minimum-window-substring)

Python ▼

⟳

```
class Solution(object):
    def minWindow(self, s, t):
        """
        :type s: str
        :type t: str
        :rtype: str
        """
        hash = [-1 for i in range(128)]
        hash1= [0 for i in range(128)]
        hash2= [0 for i in range(128)]
        length = len(t)
```

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
        for i in range(length):
            k=ord(t[i])
            hash[k]=0
            hash1[k]+=1
            hash2[k]+=1
        start=0;minstart=0;minend=2147483647
        for i in range(len(s)):
            k=ord(s[i])
            if hash[k]!=-1:
                if hash1[k]>0:
                    hash1[k]-=1
                    length-=1
                hash[k]+=1
                k=ord(s[start])
                while hash[k] > hash2[k] or hash[k]==-1:
                    if hash[k] > hash2[k]:hash[k]-=1
                    start+=1
                    k=ord(s[start])
                if length==0:
                    end=i
                    if minend-minstart > end-start:
                        minstart=start
                        minend=end
        if minend==2147483647:return ""
```

Custom Testcase □

Run Code Submit Solution

Submission Result: Accepted (/submissions/detail/42523220/)

More Details ❯ (/submissions/detail/42523220/)

Next challenges: (M) Minimum Size Subarray Sum (/problems/minimum-size-subarray-sum/) (H) Sliding Window Maximum (/problems/sliding-window-maximum/)

Share your acceptance!

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy