



## Minimum Window Substring

s = ddaaabbca      t = abc



## Minimum Window Substring

$s = \text{d d a a a b b c a}$        $t = \text{a b c}$

↓

a a b b c



## Minimum Window Substring

$s = \text{ddaaabbc}a$        $t = \text{abc}$

↓

bca



## Minimum Window Substring

$s = \text{ddaaabbcaa}$       $t = \text{abbac}$

bca X

a → 1 3  
b → 2 3  
c → 1 3





## Minimum Window Substring

↓  
s = d d a a a b b c a      t = a b c



## Minimum Window Substring

$s =$  ddaaabba  $t = abc$



## Minimum Window Substring

$s = d d a a a b b c a$        $t = a b c$

$i$  is above the first 'd' in 's'.  
 $j$  is below the first 'a' in 's'.

$d, -2$	
$a, 1$	
$b, 1$	
$c, 1$	

char, freq

cnt = 0



## Minimum Window Substring

$s =$  1 dd aa bb ca       $t = abc$

d, -2	
a, <del>0</del>	-2
b, <del>x</del>	<del>0</del> -1
c, <del>x</del>	0

char, freq

cnt = ~~0~~ x 2 3







$s = \text{ddaaabbc a}$        $t = \text{abc}$

$\text{minLen} = 10^9$        $s\text{Index} = -1$

for  $\{ i = 0 \rightarrow n \}$

hash[256] = {0}

for {





for  $i = 0 \rightarrow n$

hash[256] = {0}    cnt = 0

for  $j = 0 \rightarrow m$     hash[s[j]]++;

for  $j = 0 \rightarrow n$

if (hash[s[j]] > 0) cnt = cnt + 1;

hash[s[j]]--;

}





$\text{for } (j = 0 \rightarrow n)$

$\text{if } (\text{hash}[\text{s}[j]] > 0) \text{ cnt} = \text{cnt} + 1;$

$\text{hash}[\text{s}[j]] --;$

$\text{if } (\text{cnt} == m)$

$\text{if } (j - i + 1 < \text{minLen})$   
 $\text{minLen} = j - i + 1;$





```
for (j = 0 → n)
```

```
if (hash[s[j]] > 0) cnt = cnt + 1;
```

```
hash[s[j]] --;
```

```
if (cnt == m)
```

```
if (j - i + 1 < minLen)
```

```
minLen = j - i + 1;
```

```
startIndex = i;
```

```
break
```

```
}
```

```
}
```





if (j - i + 1 < minLen)

minLen = j - i + 1;

startIndex = i;

break

}

}

}

s.substr (startIndex, minLen);



$s = \text{ddaaaabbc a}$      $t = \text{abc}$

$\rightarrow \text{TC} \rightarrow O(n^2)$   
 $\text{SC} \rightarrow O(256)$

$\text{minLen} = 10^9$      $\text{SIndex} = -1$

$\text{for } i = 0 \rightarrow n$

$\text{hash}[256] = \{0\}$      $\text{cnt} = 0$

$\text{for } j = 0 \rightarrow m$      $\text{hash}[t[j]]++;$

$\text{for } j = 0 \rightarrow n$

$\text{if } (\text{hash}[s[j]] > 0) \text{ cnt} = \text{cnt} + 1;$

$\text{hash}[s[j]]--;$

$\text{if } (\text{cnt} == m)$





## Minimum Window Substring

$s = \text{dd} \textcolor{red}{a} \text{aa} \text{bbca}$       $t = \text{abc}$





$s = \text{ddaaaabbc a}$        $t = \text{abc}$

$TC \rightarrow O(n^2)$   
 $SC \rightarrow O(256)$

$minLen = 10^9$        $sIndex = -1$

$\text{for } \{ i = 0 \rightarrow n \}$

$\text{hash}[256] = \{0\}$        $cnt = 0$

$\text{for } \{ j = 0 \rightarrow m \}$        $\text{hash}[t[j]]++;$

$\text{for } \{ j = 0 \rightarrow n \}$

$\text{if } (\text{hash}[s[j]] > 0) \text{ cnt} = \text{cnt} + 1;$

$\text{hash}[s[j]]--;$

$\text{if } (\text{cnt} == m)$





## Minimum Window Substring

$s = \text{d d a a a b b c a}$       $t = \text{a b c}$   
l  
r

a	1
b	1
c	1

char, freq

## Minimum Window Substring

$s = \text{d d a a a b b c a}$      $t = \text{a b c}$   
     $l$              $r$

$\text{cnt} = 0$

$\text{minLen} = 10^9$   
 $\text{startIndex} = -1$

a	1
b	1
c	1

char, freq



## Minimum Window Substring

$s = d d \boxed{a} a b b c a$        $t = a b c$   
           $l$      $r$

$cnt = 0$

$minLen = 10^9$   
 $sIndex = -1$

$d, -2$

$a, 1$
$b, 1$
$c, 1$

char, freq



## Minimum Window Substring

$s = \underset{l}{dd} \underset{r}{aaa} bbca \quad t = abc$

$cnt = 0 \neq 3$

$minlen = 8$

$sIndex = 0$

d, -2	
a, 0	$r-2$
b, 0	-1
c, 0	

char, freq





## Minimum Window Substring

$s = d d \overset{x x x x x}{a a} b b c a$        $t = a b c$

$$\text{cnt} = \emptyset + x \neq y \neq z$$

min len = 8 7 6 8 4 3

$$\leq \text{Inden} = \emptyset \neq \mathbb{Z} \neq \mathbb{Z} \neq \mathbb{Z} \neq \mathbb{Z}$$

d, 0	
a, 0	$x = x + 0 \quad x \quad 0$
b, 0	$x \quad 0$
c, 0	

Chan, Jace





min len = 8 7 6 5 4 3  
startIndex = 0 1 2 3 4 5

char, freq



## Minimum Window Substring

~~xxxxxxx~~  
 $s = ddaaabbca$       $t = abc$   
                   ~~d x x~~

$cnt = 0 \ 1 \ 2 \ 2 \ 1 \ 2$

~~minLen = 8 7 6 5 4 3~~  
~~startIndex = 0 1 2 3 4 5~~

d, 0	
a, 0	<del>1 2 1 0 1 0</del>
b, 0	<del>1 0 1</del>
c, 0	

char, freq







func(s, t)  
{

n = s.size    m = t.size

hash[256] = {0}    l = 0    r = 0    maxlen = 10<sup>9</sup>

for(i = 0 → m)

start = -1

hash[t[i]]++;

cnt = 0



```
for (i = 0 → m)
    hash[t[i]]++;
    sIndex = -1
    cnt = 0
```

```
while (r < s.size())
{
    if (hash[s[s]] > 0) cnt = cnt + 1;
    hash[s[s]]--;

    (cnt == m)
    {
        if (r - l + 1 < minLen)
            m
```





```
if (hash[s[s]] > 0) cnt = cnt + 1;  
hash[s[s]] --;
```

```
(cnt == m)
```

```
{
```

```
    if (r - l + 1 < maxlen)  
        maxlen = r - l + 1  
        sIndex = l
```





```
if (r - l + 1 < minLen)
    minLen = r - l + 1
    sIndex = l
```

```
hash[s[l]] --
if (hash[s[l]] > 0) cnt = cnt - 1;
}
```



```
while (r < s.size())
```

```
{
```

```
    if (hash[s[s]] > 0) cnt = cnt + 1;  
    hash[s[s]] --;
```

```
    while (cnt == m)
```

```
    {
```

```
        if (r - l + 1 < minLen)  
            minLen = r - l + 1  
            sIndex = l
```

```
        hash[s[l]] --
```

```
        if (hash[s[l]] > 0) cnt = cnt - 1;
```

```
    }
```

```
    r = r + 1;
```





minLen = n - 1 + 1

sIndex = l

hash[s[l]] --

if (hash[s[l]] > 0) cnt = cnt - 1;

}

n = n + 1;

}

return sIndex == -1 ? ""





minLen = n - 1 + 1

sIndex = l

hash[s[l]] --

if (hash[s[l]] > 0) cnt = cnt - 1;

}

n = n + 1;

}

return sIndex == -1 ? "" : s.substr(sIndex, minLen)



```

func(s, t)
{
    n = s.size    m = t.size
    hash[256] = {0}    l = 0    r = 0    minLen = 109
    for(i = 0 → m)    → O(m)    sIndex = -1
        hash[t[i]]++;    cnt = 0

    while(r < s.size())    → O(n)
    {
        if(hash[s[r]] > 0) cnt = cnt + 1;
        hash[s[r]]--;

        while(cnt == m)    → O(n)
        {
            if(r - l + 1 < minLen)
                minLen = r - l + 1;
            sIndex = l;
        }
    }
}

```

TC







func (s, t)

n = s.size m = t.size

hash[256] = {0} l = 0 r = 0 maxlen = 109

for (i = 0 → m) → O(m) sIndex = -1  
hash[t[i]]++; cnt = 0

while (r < s.size()) → O(n)  
{

TC → O(2n) + O(m) if (hash[s[r]] > 0) cnt = cnt + 1;  
SC → O(256) hash[s[r]]--;

while (cnt == m) → O(n)  
{

if (r - l + 1 < maxlen)  
maxlen = r - l + 1  
sIndex = l

hash[s[l]]--





func l = , v

$n = s.size$     $m = t.size$

$hash[256] = \{0\}$     $l = 0$     $r = 0$     $minLen = 10^9$

for ( $i = 0 \rightarrow m$ )  $\rightarrow O(m)$     $startIndex = -1$   
     $hash[t[i]]++;$     $cnt = 0$

while ( $n < s.size()$ )  $\rightarrow O(n)$   
{

TC  $\rightarrow O(2n) + O(m)$  if ( $hash[s[s[i]]] > 0$ )  $cnt = cnt + 1;$   
SC  $\rightarrow O(256)$     $hash[s[s[i]]]--;$

while ( $cnt == m$ )  $\rightarrow O(n)$   
{

    if ( $n - l + 1 < minLen$ )  
         $minLen = n - l + 1$   
         $startIndex = l$

$hash[s[l]]--$   
    if ( $hash[s[l]] > 0$ )  $cnt = cnt - 1;$

