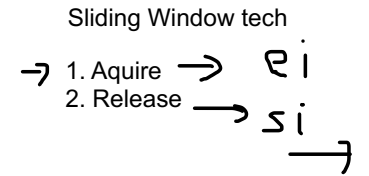


substrings-of-size-three-with-distinct-
characters/

3



during the release record the ans

ans = 12
countUniqueEleInWindow = 2
3
2 3

si
↓
0 1 2 3 4 5 6 7 8 9 10
a a b c a b c a b c a
↑
ei

26 size const arrr --> represent char with its count at some
particular window of size 3

~~0 = 7 10~~
~~1 = 4 0~~
[- |
0 - 2

size of curr Window = (ei - si + 1)

Acquire --> size of curr Window < 3 --> ei++

Release --> if(countUniqueEleInWindow == 3) --> ans++

si
↓

0 1 2 3 4 5 6 7 8 9 10
a a b c a b c a b c a

↑
ei

acquire till window size == 3

```
while(ei-si+1 < 3) {  
    acquire only  
}
```

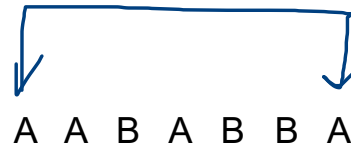
```
if (ei - si + 1 == 3) {  
    countUniq == 3 --> ans++  
}
```

```
while(ei - si + 1 >= 3) {  
    release only  
}
```

Longest Repeating Character Replacement

at most k --> 1, 2, 3, k

A A B A B B A



A = 4 B (k = 4)

B = 3 A (k = 3)

focus on mostFreqChar in curr Window -- as we need to change other chars which can be possible in least value of k

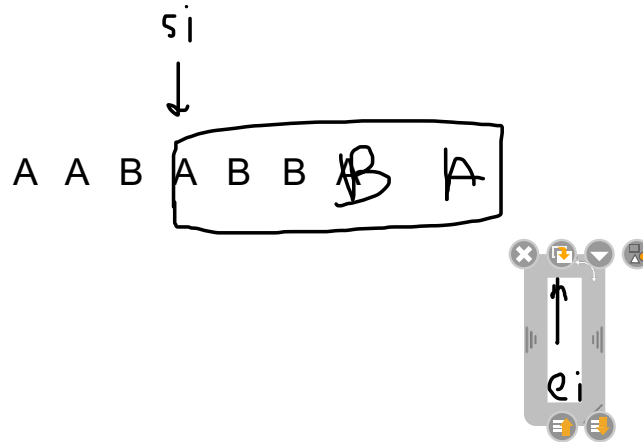
mostFreqCH = ~~2~~ 3 4

maxLen = ~~1~~ 2 3 4 5

Aquire --> everytime

Release --> when k is out of value

└─┬─┘



6 - 1

requiredK = currWindowLen - freqMostFreqChar

A - ~~1~~ 2 3 2 1 2

B - ~~1~~ 2 3 4 3