

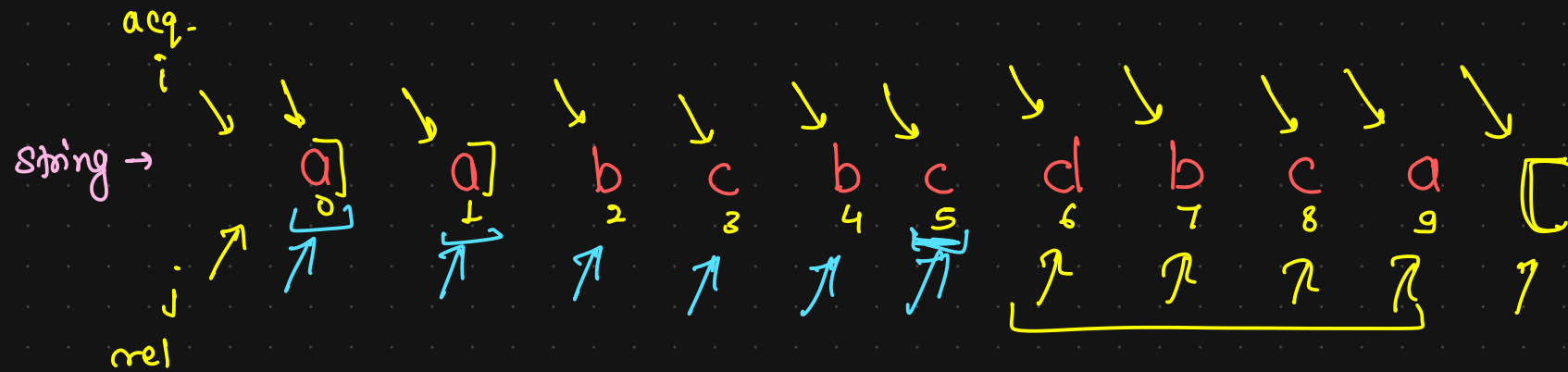
Date: 7th January 2022

Agenda

- ✓✓ Count Of Substrings Having All Unique Characters
- ✓✓ Longest Substring With Exactly K Unique Characters
- ✓✓ Count Of Substrings With Exactly K Unique Characters

Count Of Substrings Having All Unique Characters

without Repeating characters.



HashMap (Char, int)

a → 1 2 3 4 5
b → 1 2 3 4 5
c → 1 2 3 4 5
d → 1

[a]
[a, b]
[a, b, c]
[b, c]
[a, b, c]

[b]
[c, b]
[c]
[b, c]
[d, c]
[c, d]
[b, c, d]

[b]
[d, b]
[c, d, b]
[c]
[b, c]
[d, b, c]

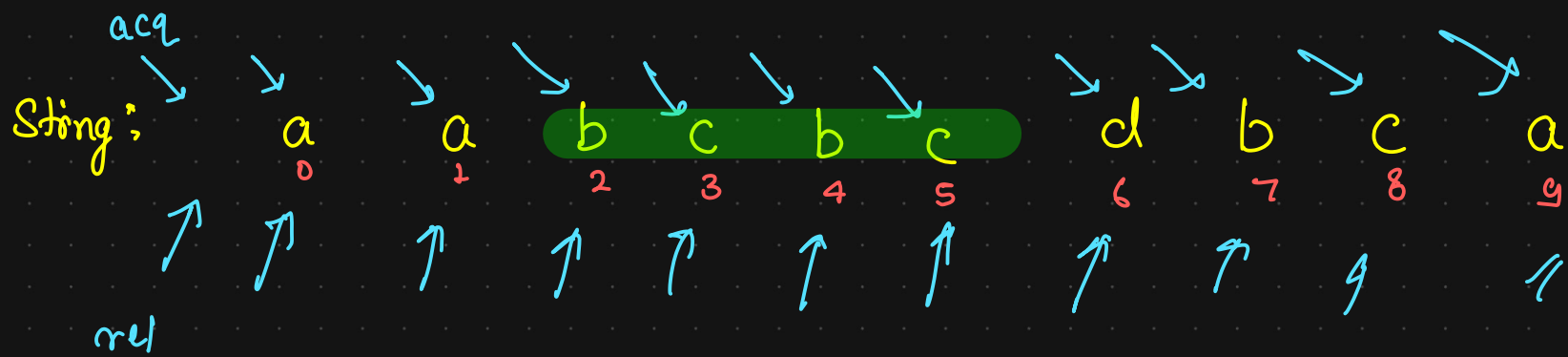
[a]
[c, a]
[b, c, a]
[d, b, c, a]

24

Longest Substring With Exactly K Unique Characters

 $k=2$

अर्थ


$$\text{Cms} = \phi / 2 \quad 4$$

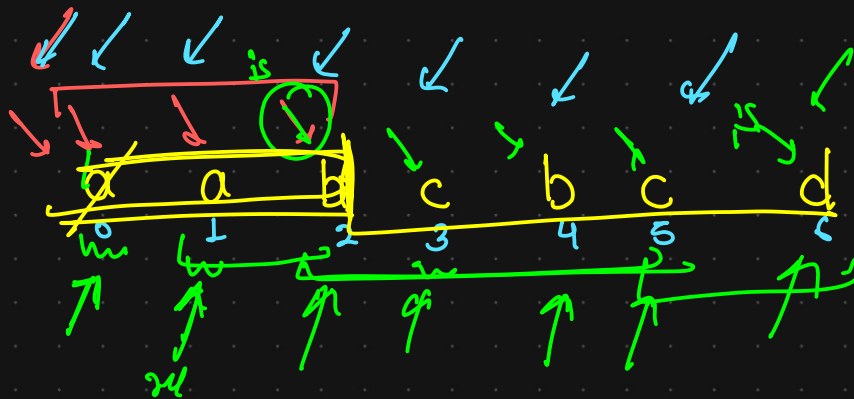
$$len = acq - tel.$$

bcb c

$$\begin{aligned} & \cancel{a + 1210} \\ & \cancel{b + 1210} \\ & \cancel{c + 1210} \\ & \cancel{d + 10} \\ & \cancel{b + 10} \\ & \cancel{c + 10} \\ & \cancel{a + 10} \end{aligned}$$

Count Of Substrings With Exactly K Unique Characters

acquire
big-
small
string =



j →
rel.

✓ k-1
a → 1 2 3 0
b → 1 2 3 0
c → 1 2 3 0
d → 1 2 3 0
b → 1 2 3 0
c → 1 2 3 0

✓ k
a → 1 2 3 0
b → 1 2 3 0
c → 1 2 3 1
d → 1 2 3 0
a → 1 2 3 0

ib
is
j

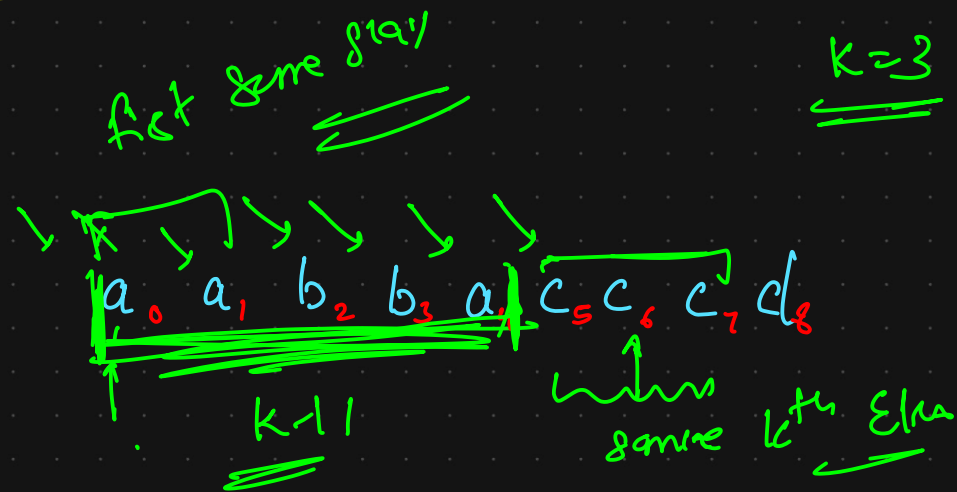
big Hashmap + k
small Hashmap + 1
delete

~~ed b~~ ~~db c~~
~~cd bc~~ ~~bca~~

$$\text{count} += \text{ib} - \text{is}$$

3
a a b c
a a b c b
a a b c b c
b c b c d
b c b c d b
b c b c d b c
a b c
a b c b
a b c b c
e b c d
e b c d b
e b c d b c
b c d
b c d b
b c d b c

Manage special case for $k=1$



$a \rightarrow \checkmark 2, 3$

$b \rightarrow \checkmark 2$

$c \rightarrow \checkmark$

$a_0, a_1, b_2, b_3, a_4, c_5$

hash map
requirement

```

int acq = -1;
int rel = -1;
int count = 0;
HashMap<Character, Integer> map = new HashMap<>();
while(true) {
    boolean flag1 = false;
    boolean flag2 = false;

    while(acq < str.length() - 1) {
        flag1 = true;
        acq++;
        char ch = str.charAt(acq);
        map.put(ch, map.getOrDefault(ch, 0) + 1);

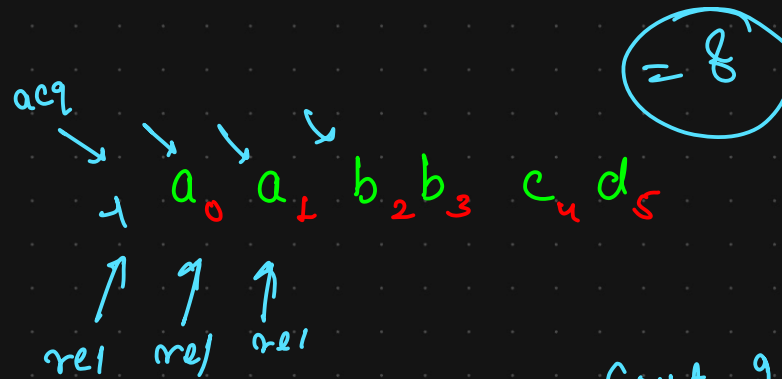
        if(map.size() == 2) {
            map.remove(ch);
            acq--;
            break;
        }
    }

    while(rel < acq) {
        count += acq - rel;
        rel++;
        char ch = str.charAt(rel);
        map.put(ch, map.get(ch) - 1);
        if(map.get(ch) == 0) map.remove(ch);

        if(map.size() == 0) break;
    }

    if(flag1 == false && flag2 == false) {
        break;
    }
}

```



Handwritten notes showing the removal of characters from the window:

~~a~~ → ~~a~~ → ~~b~~ → ~~b~~ → ~~c~~ → ~~d~~

b e 1

Handwritten notes showing the calculation of the count:

Count = 2 + 1 + 2 + 1 + 1 + 1

Diagram showing the sequence of characters and their counts:

a₀ a₁ b₂ b₃ c₄ d₅

Count = 2 + 1 + 2 + 1 + 1 + 1