

3. Longest Substring Without Repeating Characters

Medium

👍 21755

💬 971

♡ Add to List

🔖 Share

Given a string `s`, find the length of the **longest substring** without repeating characters.

`s: a b a c d a b n c b d`

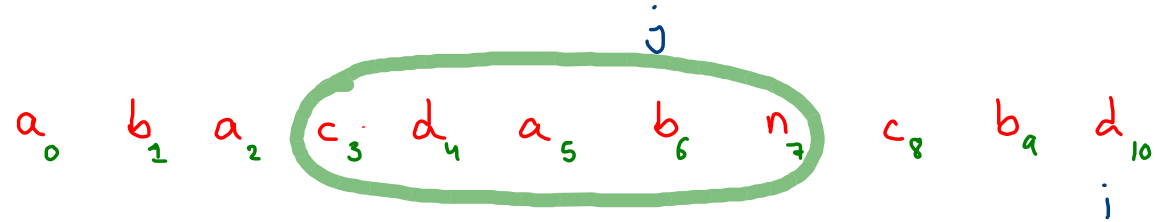
acquire : while valid acquire, stop at invalidity

release : while invalid release, stop at valid

1. acquire & release
2. hashmap

acquire : while valid acquire, stop at invalidity

release : while invalid release, stop at valid



map : $d-1$ $n-1$
 $b-1$
 $c-1$

$0len = \emptyset \neq \neq \neq \neq \neq 5$

```

while(i < s.length()-1) {
    //acquire
    while(i < s.length()-1) {
        i++;

        char ch = s.charAt(i);
        int nf = map.getOrDefault(ch,0) + 1;
        map.put(ch,nf);

        if(nf == 2) {
            //invalid
            break;
        }
        else {
            //valid
            int len = i - j; //ans -> j+1 to i
            olen = Math.max(olen,len);
        }
    }

    //release -> to be valid again
    while(j < i) {
        j++;

        char ch = s.charAt(j);

        if(map.get(ch) == 1) {
            map.remove(ch);
        }
        else {
            map.put(ch,1);
            //you are valid again
            break;
        }
    }
}

```

olen = 0 ≠ ≠ 3

a_0 b_1 a_2 c_3 b_4 c_5 a_6 a_7 t_8
↑
j
i

map: $a-1$
 $t-1$

Count Of Substrings Having All Unique Characters

 j
 i
a b c b
0 1 2 3
 i

map

b - 1

c - 1

count = 1 + 2 + 3 + 2
 a ab abc cb
 b bc b
 c c

```

while(i < str.length()-1) {
    //acquire
    while(i < str.length()-1) {
        i++;

        char ch = str.charAt(i);
        int nf = map.getOrDefault(ch,0) + 1;
        map.put(ch,nf);

        if(nf == 2) {
            //invalid
            break;
        }
        else {
            count += (i - j);
        }
    }

    //release
    while(j < i) {
        j++;

        char ch = str.charAt(j);

        if(map.get(ch) == 1) {
            map.remove(ch);
        }
        else {
            map.put(ch,1);
            //you are valid again
            count += (i - j);
            break;
        }
    }
}

```

$$\text{count} = 1 + 1 + 2 + 3 + 2 + 2 + 3 + 3 + 3 + 4$$

a_0	a_1	b_2	c_3	b_4	c_5	d_6	b_7	c_8	a_9
a	a	ab	abc	cb	bc	bcd	cdb	dbc	dbca
		b	bc	b	c	cd	db	bc	bca
			c			d	b	c	ca
									a

map

a-1
 b-1
 c-1
 d-1

Longest K unique characters substring

1. acquire: while ($\text{map.size()} \leq k$), stop
when $\text{map.size()} == k+1$ (invalid)

2. release: while ($\text{map.size()} > k$), stop
when you are valid again ($\text{map.size()} == k$)

$k = 3$

a a b a c b e b e b e

ans: c b e b e b e

1. acquire: while ($\text{map.size()} \leq k$), stop
when $\text{map.size()} == k+1$ (invalid)

2. release: while ($\text{map.size()} > k$), stop
when you are valid again ($\text{map.size()} == k$)

$\text{olen} = \cancel{0} \cancel{3} \cancel{6} 7$

$k = 3$

$\overset{j}{a_0} a_1 b_2 a_3 c_4 b_5 e_6 b_7 e_8 b_9 e_{10}$
 i

map

$e-3$

$b-3$

$c-1$

