

Longest Substring Without Repeating Characters

① abcd ecbcd

② ababbc

$O(N^2)$

len = ~~4~~ 3 a b a b c

× 1 → (a) (b) (a) (b) (c)

✓ 2 → (ab) (ba) (ab) (bc)

✓ 3 → (aba)^{*} (bab)^{*} (abc)[✓]

✓ 4 → (abab)^{*} (babc)^{*}

5 → (ababc)^{*}

$$O(N^3) \rightarrow O(N^2)$$

$$\text{len} = \cancel{0} \cancel{1} \cancel{2} \underline{3}$$

\overline{a}	b	\downarrow	\downarrow	\downarrow	c
<u>a</u>	b	a	b	c	x
	<u>b</u>	a	b	c	x
		<u>a</u>	b	c	✓
			<u>b</u>	c	
				<u>c</u>	

$$O(N^2) \downarrow \downarrow O(N)$$

a b a b c

len: ~~0~~ ~~1~~ ~~2~~ 3

first

str: - m n o p q o n q m p r

len: ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

sec

m n o p q r



S[0]
↑

'o', 'o'

256

$O(N)$
256

```
vector<bool> count(256, 0)
int first = 0, second = 0, length = 0;
while(second < s.size()) {
    while(count[s[second]]) {
        count[s[first]] = 0;
        first++;
    }
    count[s[second]] = 1;
    len = max(len, sec - first + 1);
    second++;
}
return length;
```

sec - first

$O(N)$
 $O(256)$
 $O(1)$

Smallest distinct window

Input: s = "AABBBCBBAC"

Output: 3

Explanation: Sub-string -> "BAC"

OWN

③ AAB ADB BBD BBC BCB CBA BBA
BAL ✓
BAL
length = 3

Str - A A D D B C B R A C

len = 10 6 8 4 3

$\sqrt{O(N)}$

$$\begin{aligned} \text{diff} &= 3 \quad A \\ &= 2 \quad B \\ &= 1 \quad C \\ &= \underline{\underline{0}} \quad \cancel{D} \quad \cancel{E} \quad \frac{0}{1} \end{aligned}$$

count =

			A	B	C			
0	0	0	1	0	1	0	0	0

 256

