$O(n)$ ⇒

(*)

Longest
Prefix
Suffix

	0	1	2	3	4	5	6	7	8
	A	B	C	D	E	A	B	C	D

LPS → | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 3 | 4 |

Answer

Problem
Complex

A B   A

②   o 1   ②   A   ②

A C B A C   S

φ
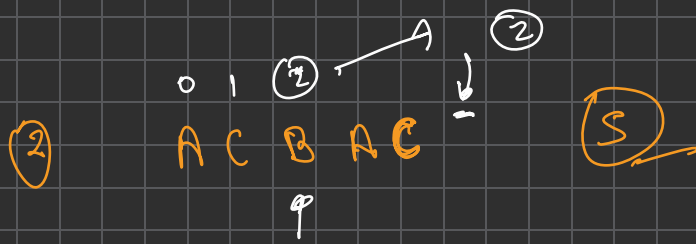
	0	1	2	3	4	5	6	7	8	9	10	11
S	a	b	c	a	b	d	a	b	c	a	b	c

LPS →

| 0 | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 4 | 5 | 3 |

→ Prefix [a b c a b] d

Suffix [a b c a b] c

S1 → a b c a b

→ S2 → a b c a b

① LPS: prefix S1 to suffix S2

S1 =
LPS

a a a u a c

Prefix a

Suffix : 1, 2, 3, 4

prefix = LPS[prefix-1]   prefix   suffix

$$\text{prefix} = 2$$

	A	B	C	A	B	D	A	B	C	A	B	D	A	B	D	A	@
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	6

(Y) ... ②

LPS | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 0 | 1 | 2 |

prefix → A B C A B D A B C
suffix → A B C A B D A B D

· LPS (m, 0)
prefix = 0, suffix = 1
while (suffix < n)
{
    if (S[prefix] == S[suffix])
    {
        LPS[suffix] = prefix+1;
        prefix++, suffix++;
    }
    else
    {
        if (prefix == 0)
        {   suffix++;

else
{
    prefix = LPS[prefix-1];
}

return LPS[n-1];

*   a a b a a b a a a

aacecaaaa

ROORSP

① S1 ⇒ "coder army strike is great"

10 1' 12 13 14 15 16

1:6 - 6

10

S2 ⇒ "strike"

O(n²)

Prefix & Prefix

LPS

	0	1	2	3	4	5
	S	t	r	i	k	e

0	0	0	0	0	0

pod

& cover

Strike coder ar my strike is great

0	0	0	0	0

LPS

✱

Learning

Combine

Strike Coder Army Strike is great

LPS

nikalne

answer

KMP:

S1 a b c d a a b c e a a b c e a a b d o p

S2 a a b c e a a b d o