Todays Content

- → Calculate lps[]
- → Optimization lps[]

Ex1:

	0	1	2	3	4	5	6	7
S =	c	a	c	y	c	a	c	a
lps[ ] =	0	0	1	0	1	2	3	2

↳ To Calculate lps[] TC: $O(N^3) \longrightarrow O(N)$?

// Calculating lps[] =

Obs1: Given S of len N & assume lps[i] = 5

S_N : S_0 S_1 S_2 S_3 S_4 ...... S_{i-5} S_{i-4} S_{i-3} S_{i-2} S_{i-1} S_i .... S_{N-1}

lps[] : _ _ _ _ _ _ _ _ _ _ 5

lps[i] = 5

$\underbrace{S_0 S_1 S_2 S_3} \cancel{S_4} = \underbrace{S_{i-4} S_{i-3} S_{i-2} S_{i-1}} \cancel{S_i}$

$S_0 S_1 S_2 S_3 = S_{i-4} S_{i-3} S_{i-2} S_{i-1}$

lps[i-1] >= 4

Is this possible? may be ✓

$S_0 S_1 S_2 S_3 S_4 == S_{i-5} S_{i-4} S_{i-3} S_{i-2} S_{i-1}$ lps[i-1] = 5

// Generalization:

lps[i] = x

lps[i-1] >= $x-1$

lps[i-1] >= lps[i] - 1

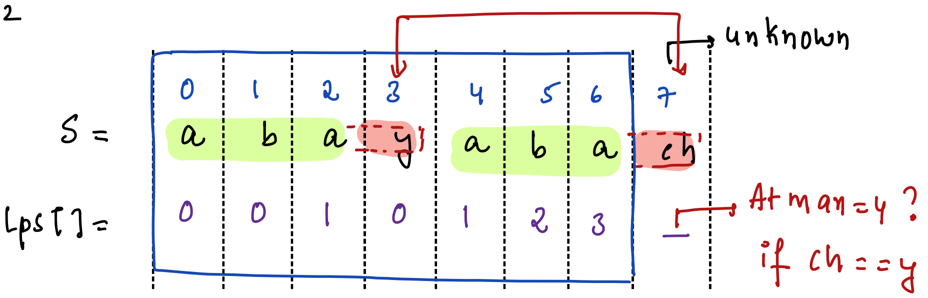
lps[i-1] + 1 >= lps[i]

$\boxed{\text{lps[i]} <= \text{lps[i-1]} + 1}$

Col1: lps[i] will at max increase by 1, compared to lps[i-1]

Step: 2

Ex1:

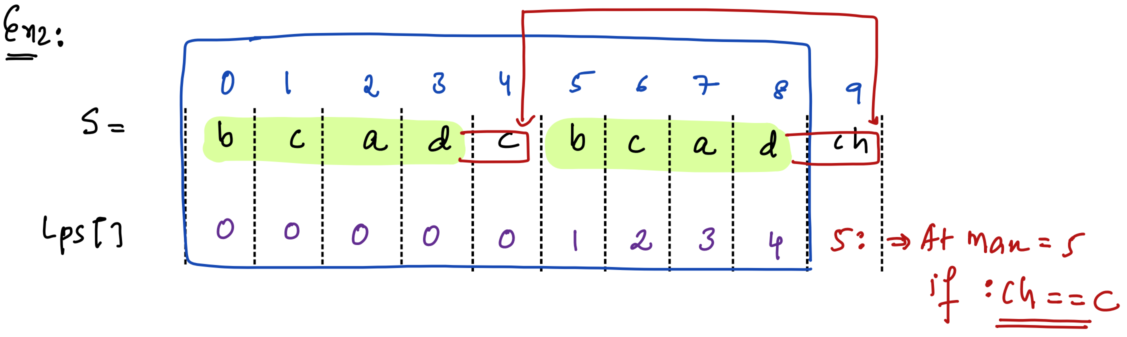


	0	1	2	3	4	5	6	7
S =	a	b	a	y	a	b	a	ch
Lps[] =	0	0	1	0	1	2	3	_

At max = 4 ?

if ch == y

Ex2:

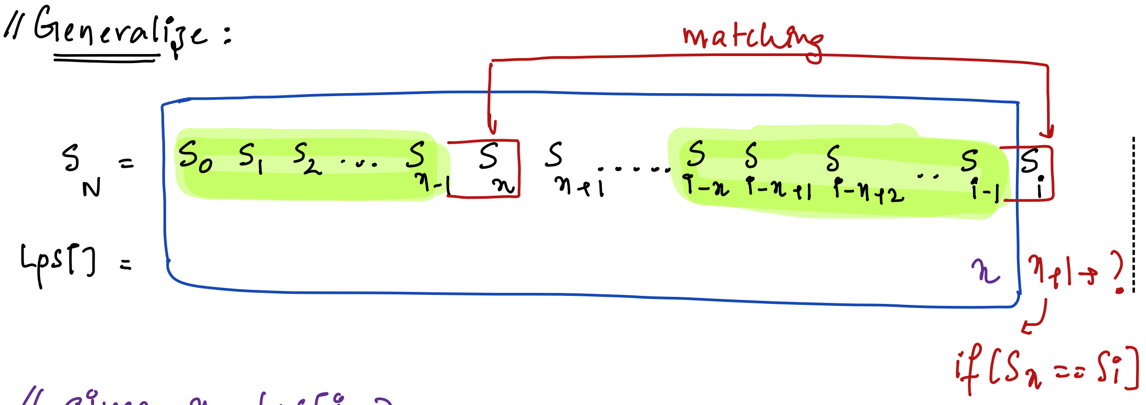


	0	1	2	3	4	5	6	7	8	9
S =	b	c	a	d	c	b	c	a	d	ch
Lps[]	0	0	0	0	0	1	2	3	4	5:

⇒ At max = 5

if : ch == c

// Generalize :



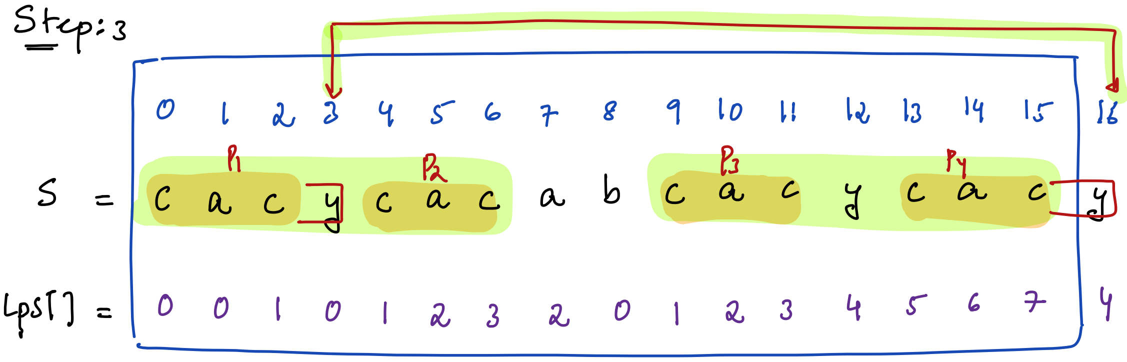
S_N = $S_0\ S_1\ S_2 \ldots S_{n-1}\ S_n\ S_{n+1} \ldots\ldots S_{i-n}\ S_{i-n+1}\ S_{i-n+2} \ldots S_{i-1}\ S_i$

Lps[] = n $n+1 \rightarrow$?

if ($S_n == S_i$)

```
// given n = Lps[i-1]
  if ( S[i] == S[n]) {
  |  Lps[i] = n+1
  }
```

Step: 3

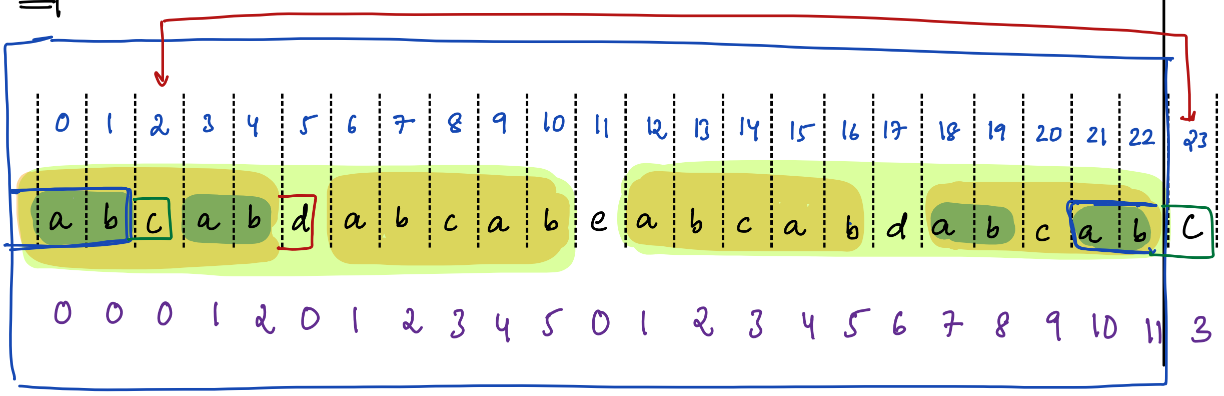


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S =	c	a	c	y	c	a	c	a	b	c	a	c	y	c	a	c	y
Lps[] =	0	0	1	0	1	2	3	2	0	1	2	3	4	5	6	7	4

i = 16, n = Lps[i-1] = 7

n	S[i] == S[n]	
7	S[16] == S[7]	* not n = Lps[n-1] n = Lps[6] = 3
3	S[16] == S[3]	Lps[i] = n+1, Lps[16] = 3+1 = 4

Step4:

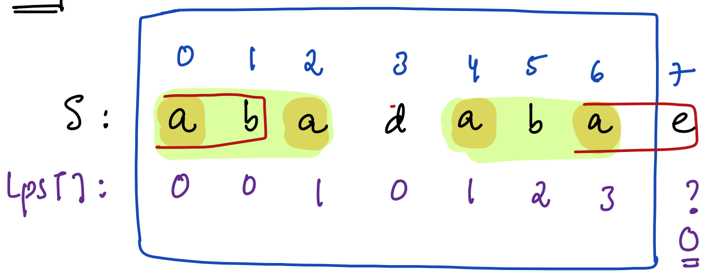


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
a	b	c	a	b	d	a	b	c	a	b	e	a	b	c	a	b	d	a	b	c	a	b	c
0	0	0	1	2	0	1	2	3	4	5	0	1	2	3	4	5	6	7	8	9	10	11	3

i = 23, n = lps[i-1] = 11

n	S[i] == S[n]	
11	S[23] == S[11]	* not n = lps[n-1] n = lps[10] = 5
5	S[23] == S[5]	* not n = lps[n-1] n = lps[4] = 2
2	S[23] == S[2]	lps[i] = n + 1 = 3

Step:5



	0	1	2	3	4	5	6	7
S :	a	b	a	d	a	b	a	e
lps[] :	0	0	1	0	1	2	3	? 0

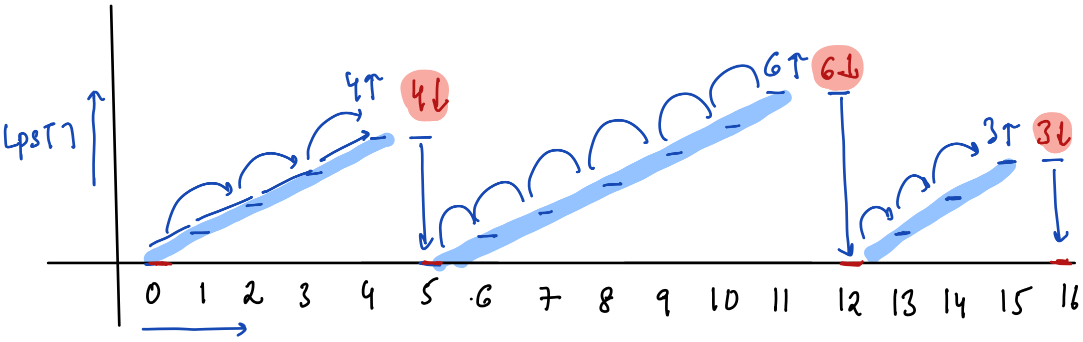
i = 7, n = lps[i-1] = 3

n	S[i] == S[n]	
3	S[3] == S[7]	* not n = lps[n-1] n = lps[2] n = 1
1	S[1] == S[7]	* not n = lps[n-1] n = lps[0] n = 0
0	S[0] == S[7]	* not n = lps[n-1] ⇏ n = lps[-1]

[if n==0, condition is not matching break]

```
int[] CreateLps(String s){
    int n= s.length
    int lps[n]
    lps[0] = 0
    i=1; i<n; i++){
        //Calalate lps[i]?
        x= lps[i-1]
        while(s[x] != s[i]){
            if(x==0){x=-1, break}
            x = lps[x-1]
        }
        lps[i] = x+1
    }
    return lps[]
}
```

Obs: In lps[i] increases by 1, it will have only 1 iteration.



Obs1: Total inc iterations = N
↳ Total dec iterations = N

TC: $O(N+N) = O(2N)$
SC: $O(N)$

Ex:

	0	1	2	3	4	5	6
	a	a	a	a	a	a	d
lps[]	0	1	2	3	4	5	?

$i = 6$, $x = lps[i-1] = 5$

x	s[x] == s[i]
5	S[5] == S[6] * x = lps[x-1], x = lps[4] x = 4
4	S[4] == S[6] * x = lps[x-1], x = lps[3] x = 3
3	S[3] == S[6] * x = lps[x-1], x = lps[2] x = 2
2	S[2] == S[6] * x = lps[x-1], x = lps[1] x = 1
1	S[1] == S[6] * x = lps[x-1], x = lps[0] x = 0
0	S[0] == S[6] * x = -1 Break

$lps[i] = x+1 = 0$

→ {Reading material}

- → Rabin karp : Heavy math, code lengthy
- → z-algo : Very confusing , code lengthy
- → KMP → {Knutt Morris Patt algorithm}
  - ↳ pattern match using lps[]

→ period : Ex: a b c a b c a b : period lps N

a b c a b c a b c a :

a b c d a b c d :

Doubts :

// A= 1010 ≠ B= 0101

T= BB = 01010101

P= 1010

	0	1	2	3	4	5	6	7	8	9	10	11	12
C =	1	0	1	0	@	0	1	0	1	0	1	0	1
lps[] =	0	0	1	2	0	0	1	2	3	4	3	4	3

A = a b c d B= i j k l

BB = i j k l i j k l

i j k l ≠ a b c d

j k l i == a b c d

k l i j == a b c d

l i j k = a b c d

i j k l ≠ a b c d

if (i j k l == a b c d)

c=c+2; c=c-1

if i j k l ≠ a b c d

return ans

```
if (A == B) {
|     c=c-1 return c
}
else
|  // return c
}
```

a b c d

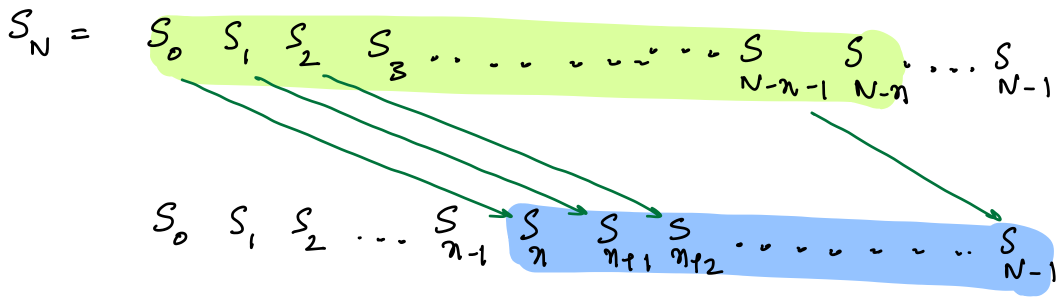
A B

Ⓐ + ← 0^{n}

Hint for period of a String:

// Give S_N, period is defined as minimum n such that

$\boxed{\forall_{i=0}^{N-n-1} S_i = S_{i+n}}$ meaning $S_0 = S_n$ $S_1 = S_{n+1} \dots S_{N-n-1}$ S_{N-1}


S_N =
S_0 S_1 S_2 S_3 . . . . S_{N-n-1} S_{N-n} . . . . S_{N-1}
S_0 S_1 S_2 . . . S_{n-1} S_n S_{n+1} S_{n+2} . . . . S_{N-1}

// if we say period of String = n,

Obs: $S[0, N-n-1] = S[n, N-1]$

Indirectly lps of S = N-n

Final obs :

	period	Lps
S_N →	n	N-n

Lps = N - period

Lps + period = N

period = N - Lps {Final conclusion}