String Pattern Matching

Aho Corasick

Rabin Karp → hashing

KMP

Z-Algo

Suffix A[]

— tree[]

— Automaton

trie

Given a string S of size N.

Prefix string : substring of S, which starts at S[0]

: S[0 — i] 0 <= i < N

Suffix string : ______ ends at S[N-1]

: S[i — N-1] 0<=i < N

S = abab

prefix string

a
ab
aba
~~abab~~

suffix strings

b ✗
ab ✓
bab ✗
~~abab~~

LPS of a string : Length of the longest prefix which is also a suffix of the string!
except the full string.

LPS(abab) = 2

S = a b c b a ⟶ LPS : 1

prefix

a
ab
abc
abcb
~~abcba~~

suffix

a ✓
ba ✗
cba ✗
bcba ✗
~~abcba~~

① LPS("a") → 0

Prefix	Suffix
~~a~~	~~a~~

② LPS("aaaaa") → 4

P	S
a	a
aa	aa
aaa	aaa
aaaa	aaaa
~~aaaaa~~	~~aaaaa~~

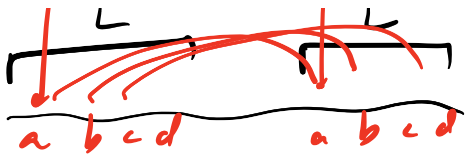
③ S = abcab

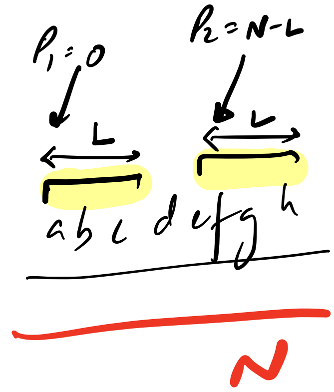
P	S	time
a	b	1
ab	ab	2
abc	cab	3
abca	bcab	4
~~abcab~~	~~abcab~~	⋮
		N-1

$$\frac{(N-1)(N)}{2}$$

$$\boxed{TC: O(N^2)}$$

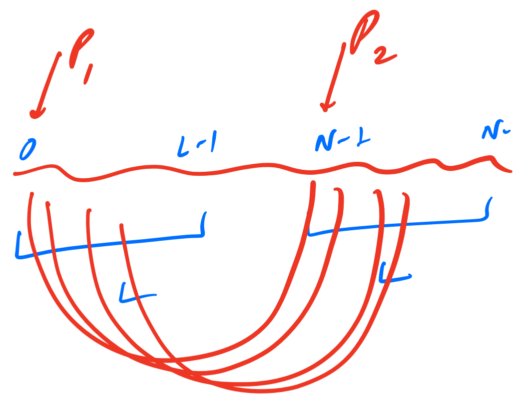
P_1 P_2






```
ANS = 0;
f(L = 1; L < N; L++){
    ok = true;
    P1 = 0, P2 = N-L;
    f(i = 0; i < L; i++){
        if(S[P1] != S[P2]){
            ok = false;
            break;
        }
        P1++, P2++;
    }
    if(ok == true)
        ANS = L;
}
ret ANS;
```

$\boxed{TC: O(N^2)}$



Q. Given a string S of length N.
Return the LPS Array!

LPS[i] = LPS value of the substring s[0…i]!

	0	1	2	3	4	5	6
S =	a	a	b	a	a	b	a
LPS[]	0	1	0	1	2	3	4

$N \times O(N^2)$

$\boxed{TC = O(N^3)}$ $\xrightarrow{KMP}$ $O(N)$

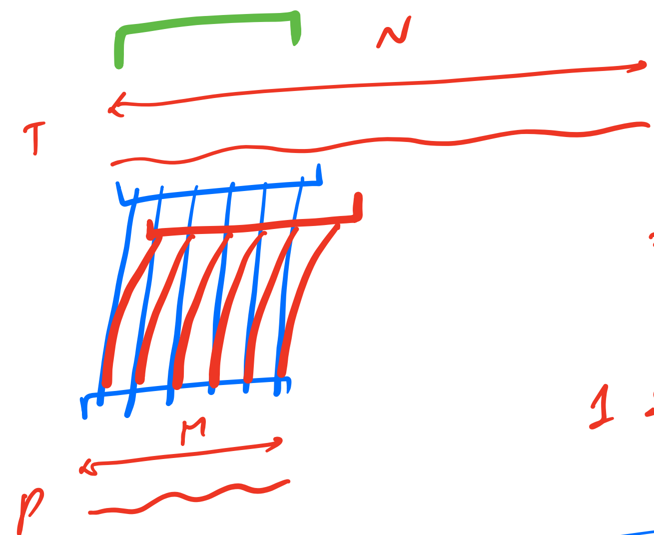
S :	a	a	b	a	c	a	a	b	a
LPS :	0	1	0	1	0	1	2	3	4

Q Given a text (T) & a pattern (P). Check if the pattern is present in the text as a substring!

length

$N \leftarrow T:$ a a $\underbrace{\text{b a c d}}$ $\longrightarrow$ true

$M \leftarrow P:$ $\underbrace{\text{a b a c}}$



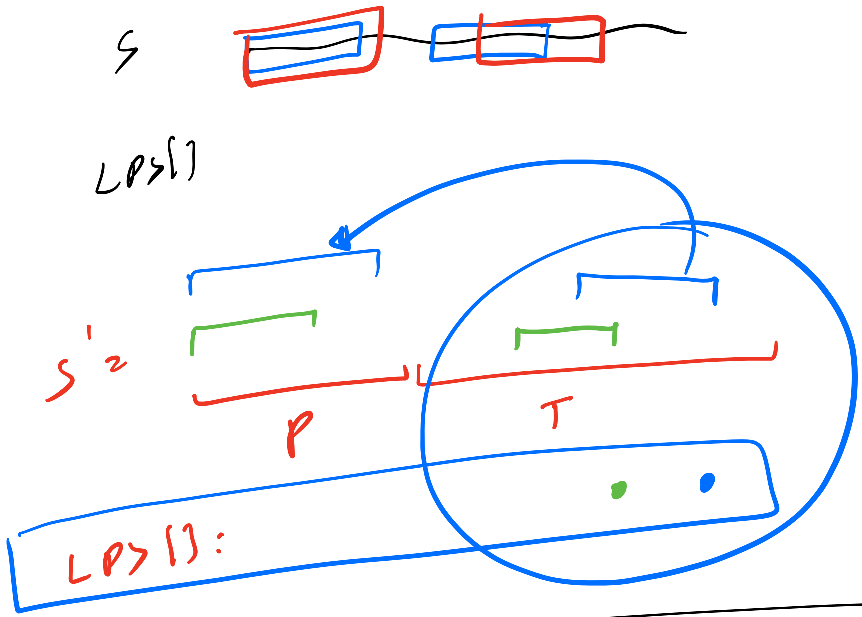
$\#SS$ of len $M \longrightarrow N-M+1$

1 SS comparision $\longrightarrow O(M)$

$\boxed{TC = O((N-M)\,M)} \simeq NM$

$\boxed{1 <= M <= N <= 10^6}$

II LPS



T: a a b a c d

P: a b a c

S' = P + T

M

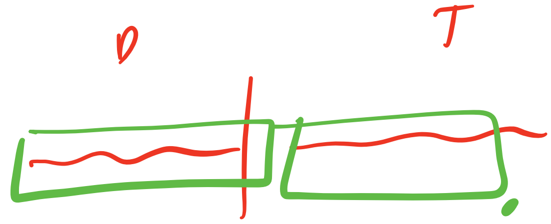
N

s' =	a	b	a	c	a	a	b	a	c	d
LPS[]:	0	0	1	0	1	1	2	3	4	0

$\to$ M

T: a $\quad N = 1$

P: a a $\quad M = 2$



$S' = P + T$

a a a

LPS[]: 0 1 2

M

Idea: $S' = P + @ + T$

delimiter

[Not be in P or T]

$S' = \overbrace{a\ a}^{2}\ @\ a$

LPS[]: 0 1 0 1

X

T: a, $\quad P$: a a a

$S' = P + @ + T$

a a a @ a

LPS: 0 1 2 0 1

X

$$S' = \overbrace{a \quad a}^{M} \; @ \; a \quad a$$

$$LPS[\,] : \; 0 \quad 1 \quad 0 \quad 1 \quad 2 \to M \; ✓$$

STEPS:

1. $S' = P + @ + T$ → $O(N+M)$
2. Construct $LPS[\,]$ for S' → KMP $O(N+M)$
3. find for any i
   $\text{if } (LPS[i] == M)$ → is a match
   $O(N+M)$

$$\boxed{TC = O(N+M)}$$

$$\boxed{SC = O(N+M)}$$ S', $LPS[\,]$.

Q. Given T & P. Count the no. of occurrences of P in T.

T : ababababaab → 3

P : aba

M = 3

S' = P + @ + T

a b a @ a b a b a b a b a a b

LPS[] : 0 0 1 0 1 2 3 2 3 2 3 1 2

Count the # of occ. of M

Q. Given a string S. Find out the no. of cyclic rotations of S which are equal to S.

S = a b a b

1 rot

Cyclic rotations

1 : b a b a
2 : a b a b
3 : b a b a
4 : a b a b

2

1) BF $\longrightarrow N^2$

2) $S = abcd \xrightarrow{rot}$

abcd
bcda
cdab
dabc

$s' = s + s$

$= abcdabcd$

$s' = s + s -$ lst char

P : s

T : s'

abcd

abcdabc

$s'' = P + @ + T$

$S + @ + s'$

s''	a	b	c	d	@	a	b	c	d	a	b	c
LPS:	0	0	0	0	0	1	2	3	4	1	2	3

$S = abab$

$S' = S + S - 1$: $2N-1$

$= abab\ aba$

$S'' = P + @ + T$: $3N$

$S + @ + S'$

abab @ abababa

LPS[] 0 0 1 2 0 1 2 3 4 3 4 3 → 2

N = 4

$\boxed{TC = O(N)}$

$\boxed{SC = O(N)}$