String = " abacbb"

Pattern = " bac "

Brute force = $O(MN)$

Many algorithms to solve it optimally $\simeq O(N+M)$

1) Rabin - Karp
2) KMP
3) Z- algorithm
4) etc

$O(n+m)$

1) Prefix of a string.

$S \Rightarrow$ "abcda" (0 1 2 3 4)

Prefix $\Rightarrow$ a, ab, abc, abcd and abcda

$\Rightarrow$ Substring starting from 0th index.

2) Suffix of a string.

$S =$ "abcda" (0 1 2 3 4)

Suffix = a, da, cda, bcda, abcda.

Suffix is a substring which ends at $(n-1)^{th}$ index.

3) LPS of a string $\Rightarrow$ longest prefix of a string which is also a suffix.

longest prefix suffix

$S \Rightarrow$ abcab

LPS(s) $\Rightarrow$ 2

Prefix	Suffix
a	b ✗
ab	ab ✓
abc	cab ✗
abca	bcab ✗
abcab	abcab ✗

Ex $S =$ abcd

$LPS(S) \Rightarrow 0$

Prefix	Suffix
a	d ✗
ab	cd ✗
abc	bcd ✗
abcd	abcd

Ex $S =$ aaaaa

$LPS(S) = 4$

Prefix	Suffix
a	a ✓
aa	aa ✓
aaa	aaa ✓
aaaa	aaaa ✓
aaaaa	aaaaa

Note: To calculate the lPS we don't consider the full string.

Brute Force to find lps of a string !

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ a_1 & a_2 & a_3 & a_4 \end{array}$$

$S = a_1\ a_2\ a_3\ a_4$

$l \Rightarrow 2$ $\quad P_1 \quad P_2$

$[1, n-1]$

mid

Prefix	Suffix
a_1	a_4
$a_1 a_2$	$a_3 a_4$
$a_1 a_2 a_3$	$a_2 a_3 a_4$

Pseudo Code

lps ⇒ [1, n-1]

```
int ans_lps ⇒ 0;

for (int lps ⇒ 1 ; lps < n ; lps++) {
    int p1 ⇒ 0 ; int p2 ⇒ n - lps;
    while (p1 < lps) {
        if (s[p1] == s[p2])
            p1++ , p2++;
        else
            break;
    }
    if (p1 == lps)
        ans_lps
        = lps
}
```

$S =$ "abcab"

0 1 2 3 4

P_1 P_2

lps	prefix	suffix
1	a	b
2	ab	ab
3	abc	cab
4	abca	bcab

LPS Array : Given a string of length n, lps array would also contain n elements.

where lps[i] ⇒ lps of the substring from index 0 to index i.
S[0....i]

S ⇒ a a b a a b a

lps ⇒

0	1	2	3	4	5	6
0	1	0	1	2	3	4

lps[0] ⇒ lps(S[0....0])

lps[1] ⇒ lps(S[0...1]) = lps(aa)

lps[2] ⇒ lps(S[0...2]) = lps(aab)

⇒

prefix	suffix
a	b
aa	ab

Brute force ⇒ $O(n^3)$

⇓

Optimized ⇒ $O(n)$

Q1

S ⇒ aabacd

P ⇒ abac

Find if the pattern exists in the string.

S' ⇒ P + S

0	1	2	3	4	5	6	7	8	9
a	b	a	c	a	a	b	a	c	d

0	1	2	3	4	5	6	7	8	9
0	0	1	0	1	1	2	3	4	0

if lps[i] == length of Pat

there is a pattern match

TC: O(N+M)

SC: O(N+M)

Edge Case : $S = abcd$.
$P = aa$

$S' \Rightarrow P + S$

$\Rightarrow \boxed{a\,a}\,\boxed{a}\,b\,c\,d$

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

$S' \Rightarrow P + \text{"\#"} + S$

character which is not present in either pat or main string

$S' \Rightarrow aa\#abcd$

0	1	0	1	0	0	0

$S \Rightarrow a\,a\,a\,a$

$P \Rightarrow a\,a$

$S' \Rightarrow P + S$

$a\,a\,a\,a\,a\,a$

0	1	2	3	4	5

$S' \Rightarrow P + \# + S$

$a\,a\,\#\,a\,a\,a\,a$

0	1	0	1	2	2	2

Q2 Given a string S and a pattern, find the number time pattern occurred in S.

Ex : S = ab ababaaab
P = ab

S' ⇒ P + (#) + S

S' ⇒ ab#ababaaab

0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	1	2	1	2	1	2	1	1	1	2

TC: O(n+m)
SC: O(n+m)

Q3 Given a string S. Find no of cyclic rotations of S which is equal to S.

Ex S ⇒ abcd

dabc

cdab

bcda

abcd

ans = 1

⇒ N cyclic rotations possible

S ⇒ abab

baba

abab

baba

abab

ans = 2

$S' \Rightarrow S + S$

0	1	2	3	4	5	6	7
a	b	a	b	a	b	a	b

$\Rightarrow$ It has all cyclic rotations of the original string as substring

$\Rightarrow$ Substrings of length.

1) $S[0-3] \Rightarrow$ abab
2) $S[1-4] \Rightarrow$ baba
3) $S[2-5] \Rightarrow$ abab
4) $S[3-6] \Rightarrow$ baba
5) $S[4-7] \Rightarrow$ abab

$S' \Rightarrow S + S -$ { last character }

$S' \Rightarrow$

0	1	2	3	4	5	6
a	b	a	b	a	b	a

$S'' \Rightarrow S + \text{"\#"} + S'$

abab # abababa

0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	2	0	1	2	3	4	3	4	3

Q Given a string s. Calculate the min no of characters to add in the beginning to s to make it a palindrome

Ex 1 dcabacd

Ex 2 ed aabaadef.

Ex 3 ed abbade

$L \Rightarrow$ length of the longest prefix palindrome.

ans $\Rightarrow$ $N - L$

Prefix palindrom $\Rightarrow$ Prefix which is equal to it's reverse

Ex a b a c d

a b a c d # d c a b a

3

Ex a b a b a

a b a b a # a b a b a

5

$a_1 \; a_2 \; a_3 \; a_4 \; a_3 \; \# \; a_5 \; a_4 \; a_3 \; a_2 \; a_1$

$\Downarrow$

$\times$

$a_1 = a_4$

$a_2 = a_3$

$a_3 = a_2$

$a_4 = a_1$