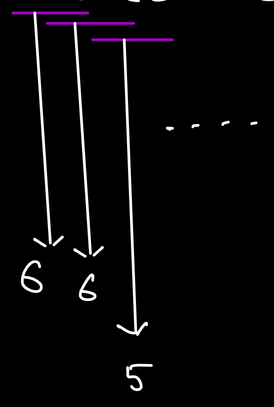
Q: Given a String of length M. Return the sum of ASCII values of every substring of length N.

S: acbabcdefabc ⇒ M

N=3

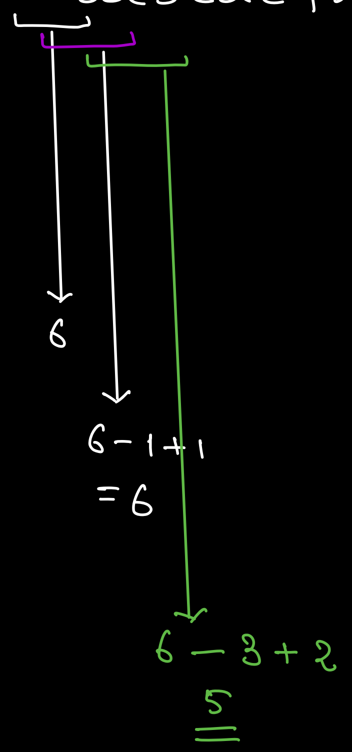
a → 1
b → 2
c → 3
d → 4



→ Return int[].

No. of substrings of length N in a String of length M = $\boxed{M-N+1}$

- - + +
acbabcdefabc



TC: O(M)
SC: O(1)

Q. Given a string (Text) of len M & a small string (pattern) of len N, Count the number of occurrences of pattern in the text.

$M >> N$.

T: abcxyclmoxycljpqrycmkrtxycl

P: xycl

Approach 1:

T: abcxyclmoxycljpqrycmkrtxycl

P: xycl

T: abababab } count = 3

P: abab

# of substrings of len = N ⇒ $M-N+1$ $(M>>N)$

$\approx M$

# of comparisons in 1 substring = N

# of iterations ⇒ $N \cdot M$

TC ⇒ $O(M \cdot N)$

SC : $O(1)$

Approach 2 : HashMap.

$$\langle \underset{\underset{\text{String}}{\uparrow}}{k}, \underset{\underset{\text{int}}{\uparrow}}{v} \rangle$$

⇒ Insert all the substrings of length N in the HashMap with their frequency.

```
for every substring (s) of length N:
        if (map. contains (s))
                map[s] ++;
        else {
                map. insert (s, 1);
```

TC of inserting a String (N) in HashMap.

Calculating HashCode of an int ⇒ O(1)

Calculating Hashcode of a String ⇒ O(N)

TC to insert a String in Map ⇒ O(N).

TC to search a String in Map ⇒ O(N)

TC to insert M strings in HM ⇒ $\underline{O(MN)}$

SC : $O(M \cdot N)$

T: abcxyclmoxycljpqxycmkrtxycl
P: xycl

[abcx, bcxy, cxyc, xycl, ------]

Array of String

Size = $\underline{M-N+1}$

P: xycl

$M >> N$
$\Rightarrow \approx M$

$$\boxed{TC : O(M * \underline{N})}$$

Array of $\underline{\text{int}}$:

[78, 81, 49, 35, 42, 21, ----] ⇒ $\underline{M}$

$k = \underline{35}$ ⇒ freq(k)

TC: $\underline{O(M)}$

String Comparison ⇒ $O(N)$
int Comparison ⇒ $O(1)$

⇒ Find sum of ASCII value of substring of length N.

T: abcabacdef   P: abc

[6, 6, 6, 4, 6, - - - -]

a → 1
b → 2
c → 3
⋮

$\underbrace{\text{Sum of ASCII value}}_{\text{Hashcode}}$ of pattern = 6

⇒ If the hashcode is NOT matching:
Strings aren't equal.

⇒ If the hashcode is matching:
⇒ Strings can be equal
⇒ Char by char by matching

Best Case : O(M) { No matches }
↑
Build the Array of hashcode of all substring.

Worst Case : TC: O(NM) { All matches }

SC: O(M) ⟶ O(1)

T: a a a a a a a

P: a a a

A: [3 3 3 3 3]

$hc(P) = 3$

1) $h(abc) = h(acb) = h(bac) = h(bca)$

$= h(cab) = h(cba) = h(aad)$

2) $h(aabc) = h(abbb)$

$123 \Rightarrow 1\times 10^2 + 2\times 10 + 3$

321

231

213

$$h(\overset{0}{a}\overset{1}{b}\overset{2}{c}) = a \times p^0 + b \times p^1 + c \times p^2$$

$$h(acb) = a \times p^0 + c \times p^1 + b \times p^2$$

$T: \underbrace{abc}\,defgh$

$T: a\underbrace{bcd}\,efgh$

$\underline{\underline{N=3}}$

$$h(abc) = a \times p^0 + b \times p^1 + c \times p^2$$

$$\Big\downarrow -a$$

$$b \times p^1 + c \times p^2$$

$$\Big\downarrow /p$$

$$b \times p^0 + c \times p^1$$

$$\Big\downarrow +d \times p^2$$

$$h(bcd) = b \times p^0 + c \times p^1 + d \times p^2$$

$\underline{\underline{N=4}}$

$T: \overset{\times}{\underline{a}}\,\overline{\underline{bcd}\,e}\,fgh$

$$h(abcd) = a \times p^0 + b \times p^1 + c \times p^2 + d \times p^3$$

$$\Big\downarrow -a$$

$$b \times p^1 + c \times p^2 + d \times p^3$$

$$\Big\downarrow /P$$

$$b \times p^0 + c \times p^1 + d \times p^2$$

$$\Big\downarrow + e \times p^3 \rightarrow \underline{\underline{N-1}}$$

$$h(bcde) = (b \times p^0 + c \times p^1 + d \times p^2 + e \times p^3) \% K$$

T: a a a a a a a a

P: a a a a

$$h(str) = \left(\sum_{i=0}^{N-1} str[i] \times p^{i} \right) \% K$$

$$h(bcde) = \left(\frac{h(abcd) - a}{P} + e \times p^{3} \right) \% K$$

⇒ Inverse <u>Modulo</u>

⇒ Fermat's Little <u>Theorem</u>.

$$(p^{n}) \% K \Rightarrow pow(p, n, K)$$

$$h(abc) = \left(\sum_{i=0}^{N-1} str[i] \times p^{i} \right) \% K \in [0, K-1]$$

⇒ **RABIN KARP** <u>Algo.</u>

⇒ $P \in (29, 31, 37 - - - -)$

✡ $\overset{0\ 1\ 2}{abcd}$

$\xrightarrow{\qquad}$

$\xleftarrow{\qquad}$ $\underline{2\ 1\ 0}$ } ⇒ $\textcircled{P}$ ×

$$\overset{2}{a}\,\overset{1}{\underline{b}}\,\overset{0}{\underline{c}}\,\underline{d}$$

$$h(abc) \quad c\times p^0 + b\times p^1 + a\times p^2$$

$$\downarrow -ap^2$$

$$c\times p^0 + b\times p^1$$

$$\downarrow *p$$

$$C\times p^1 + b\times p^2$$

$$\downarrow +d$$

$$h(\overset{2}{b}\overset{1}{c}\overset{0}{d}) \Rightarrow d\times p^0 + c\times p^1 + b\times p^2$$

HW

Given a String N.

Prefix Substring : Substring starts with Index=0

Suffix Substring : Substring ends at index = N-1.

S : abab

Prefix	Suffix
a	b
ab	ab
aba	bab
abab	abab

Quiz

S: "breakthe bias"

↓

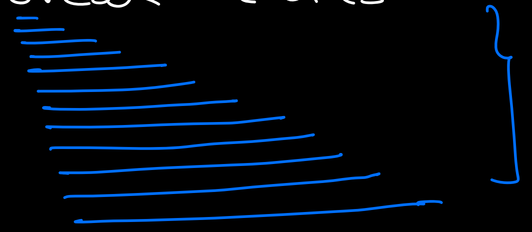
Not Prefix
Substring

Perfect Prefix : Starts at index = 0 & ends at index < N-1

Perfect Suffix : ends at index = N-1 & Starts at index > 0.

Quiz

breakthe bias



12 Prefix Substrings.

# of prefix substrings = N

# of perfect prefix substrings = N-1

Q: Given a string of length N, find the length of longest prefix that is also a suffix substring.

prefix → Perfect

suffix → Perfect

⇒ LPS → Longest Prefix that is also a Suffix.

$$\begin{array}{rl} & \scriptstyle 0\ 1\ 2\ 3\ 4 \\ S: & abcab \end{array} \Rightarrow ②$$

Prefix	Suffix
a	b
ab	ab
abc	cab
abca	bcab

Quiz

$$\begin{array}{rl} & \scriptstyle 0\ 1\ 2\ 3\ 4\ 5\ 6 \\ S: & abcdabc \end{array} \Rightarrow ③$$

Prefix	Suffix
a	c
ab	bc
abc	abc
abcd	dabc
abcda	cdabc
abcdab	bcdabc

Quiz

S: aaaaa

⇒ 4

Quiz

S: a

⇒ 0

S: $s_0\ s_1\ s_2\ s_3\ s_4\ s_5$

$s_0 s_1 s_2 s_3 s_4$	$s_1 s_2 s_3 s_4 s_5$ ⇒ N-1
$s_0 s_1 s_2 s_3$	$s_2 s_3 s_4 s_5$ ⇒ N-2
$s_0 s_1 s_2$	$s_3 s_4 s_5$ ⇒ N-3
$s_0 s_1$	$s_4 s_5$
s_0	s_5 ⇒ 1

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

Q. Given a String of length N, return the LPS[].

LPS[i]: Length of longest prefix that is also a suffix from index 0 to i

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

Quiz

	0	1	2	3	4	5	6	7	8
S:	a	a	b	a	c	a	a	b	a
LPS[]	0	1	0	1	0	1	2	3	4

TC of building LPS[] ⇒ $\boxed{O(N^3)}$

$\downarrow$

$O(N)$

KMP (Knuth Morris Prat) Algo.

—— * ——