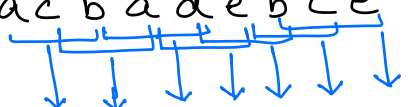



Q.1 Given a string of length M . Return the sum of ASCII values of every substring of length K .

S: "acbadebce"

 Ans: [6, 6, 7, 10, 11, 10, 10]

$K=3$.

assume
 $a \rightarrow 1$
 $b \rightarrow 2$
 $c \rightarrow 3$
 \vdots

s: acbadebce

 ascii = 6

TC: $O(M)$

SC: $O(1)$ { Excluding the Ans. }
 { array from SC }

Q.2 Given a large string (Text) of length M and a small string (pattern) of length N . Count the no. of occurrences of pattern in the text.

Text: "abcxyzlmnxyzjpxycmkxtxyz"

Pattern: "xyz"

\Rightarrow 3

1) Sliding Window $N=4$

abcxycdlmoxxycdjprxycdmkatxycd
xycd

count = (3).

TC: $O(M \cdot N)$

No. of substrings of length $N \Rightarrow M - N + 1$
($M \gg N$) $\approx M$

No. of comparison in every substring = N .

2) HashMap.

abcxycdlmoxxycdjprxycdmkatxycd

xycd $\Rightarrow \underline{N}$

* ① for every substring s of length N in text:

if (s is present in the map):

map[s]++

else

map.insert(s, 1);

② `return map[pattern];`

* Hash code calculation of int $\Rightarrow O(1)$

* Hash code calculation of String $\Rightarrow O(N)$.

* TC of insertion of a string of length N in map $\Rightarrow O(N)$.

Overall TC: $O(M \cdot N)$.

SC: $O(M)$

* abcxyclmoxycljprxyckmkatxycl

f: xycl

["abcx", "bcxy", "cxyz", "xycl", "yclm", ...]

TC: $O(MN)$

* [12, 78, 95, 1, 5, 2, 3, 5, ...]

freq(5) $\Rightarrow 2$

$\hookrightarrow O(N)$

* Integer comparison $\Rightarrow O(1)$

* String comparison $\Rightarrow O(N)$

Ex

Text: abcdabca

Pattern: abc ($N=3$) \Rightarrow 6

Sum of ASCII

$\rightarrow [6, 9, 8, 7, 6, 6]$

\rightarrow

Count = 0, \neq 2

Assume

$a \rightarrow 1$

$b \rightarrow 2$

$c \rightarrow 3$

\vdots

$z \rightarrow 26$

*

$\text{hashCode}(S) = \text{sum of ASCII values of } \underline{S}$

*

if hashCode is NOT matching :
 \Rightarrow Two strings are NOT same.

*

if hashCode is matching :
 \Rightarrow Two strings can be same
 \Rightarrow Char by char comparison.

TC:

Best Case :- $O(M)$ [No matches]

\uparrow
find hash code of every substring.

Worst case :- $O(M \cdot N)$ [All matches]

* Every substring is having the same hash code as of pattern.

S: "aaaaaa"
P: "aa" } Worst case scenario.

$$SC: O(1)$$

$$\Rightarrow h(abc) = h(acb) = h(bac) = h(bca) \\ = h(cab) = h(cba)$$

* All the anagrams has same hash code.

$$\Rightarrow h(aabc) = h(aaad)$$

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

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

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

$$\begin{aligned} 123 &\Rightarrow 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 \\ 321 &\Rightarrow 3 \times 10^2 + 2 \times 10^1 + 1 \times 10^0 \end{aligned}$$

a b c d e f g h

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

$$\downarrow -a$$

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

$$\downarrow /p$$

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

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

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

$$h(bcd) = \frac{h(abc) - a}{p} + d \times p^2$$

Text: abcdefg

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

↓ -a

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

↓ /p

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

↓ + e × p³ → N-1

$$h(bcde) = b \times p^0 + c \times p^1 + d \times p^2 + \boxed{e \times p^3}$$

Ex

Text: a a a a a a a a

Pattern: a a a

TC: $O(M \cdot N)$

(worst case)

* Issue

↳ Overflow

↳ Take modulo while finding the hash code.

$$(x) \% K \rightarrow (10^9 + 7)$$

$$\text{hash-code}(str) = \sum_{i=0}^{N-1} (str[i] \cdot K * P^i \cdot K) \cdot K.$$

$$P \in [29, 31, 37, \dots]$$

↓
Prime
No.

* Choose P from this range to reduce no. of collisions.

* Rabin Karp Algorithm.

HW :-

1) Try to implement.

2) Tweak the hash code funⁿ such that inverse mod is NOT required.

(Index the substrings from to right to left instead of left to right)

Given a string of length N .

Prefix Substring: substring starting from index = 0.

Suffix Substring: substring ending at index = $N-1$.

$s = \text{"abab"}$

Prefix substring		Suffix substring	
Perfect Prefix substring	a	b	Perfect Suffix substring.
	ab	ab	
	aba	bab	
	abab	abab	

Quiz

$s: \text{"breakthebias"}$

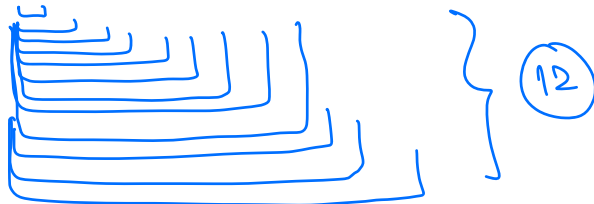
Prefix substring.

Perfect prefix substring :- Starts from index = 0
& ends at index $< N-1$

Perfect suffix substring :- Ends at index = $N-1$
& starts at index > 0 .

Quiz-2

$S = \text{"breakthebias"}$



$\text{start_index} = 0$

$\text{end_index} \in [0, N-1] \Rightarrow \underline{\underline{N}}$

of Prefix Substrings = N .

of Suffix Substrings = $\underline{\underline{N-1}}$

Q. Given a string. Find the length of largest prefix substring which is also a suffix substring.

$S = \text{"abcab"}$

PS

a
ab
abc
abca

SS

b
ab
cab
bcab

$\Rightarrow \underline{\underline{(2)}}$

Quiz

$S = \text{"abcdabc"}$

PS

a
ab
abc
abcd
abcda
abcdab

SS

c
bc
abc
dabc
cdabc
bcdabc

$\Rightarrow \underline{\underline{(3)}}$