Q Given 2 strings A & B.

Check if they are anagrams. !!

Anagrams: If 2 strings A & B are said to be anagrams if they are permutations of each other.

	A	B	
Ex	tac	cat	✓
	ayush	sayud	✗
	madam	dadam	✗

Soln 1) Sort both the strings. & compare

T.C. = N log(N)

S.C. = O(1)

2) If Distinct chars

⇒ Take XOR of all chars of A & B

if 0, return true

else return false.

A:

a a b b c

B:

c e f f c

3) HashMap

Small Case = O(26)

Small + Cap = O(52)

Create 2 HashMaps (hm1 ≠ hm2)

hm1 → A

hm2 → B

for (char a → z) {

T.C. = O(N + N + 26/52)

S.C. = O(1)

	97	98	99
A =	a	b	c
B =	b	c	a
	98	99	97

a ^ b ^ c ^ b ^ c ^ a

97 ^ 98 ^ 99 ^ 98 ^ 99 ^ 97 = 0

Q Given 2 strings A & B. (N>=M)

N M

Count the no. of substrings of A which are anagrams of B.

Eg: A: a b c a c c b a b c a c

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

N = 12

B: a b a c c

0 1 2 3 4

M = 5

Sliding window of size M

	A	B
0-4	a b c a c	✓ a b a c c
1-5	b c a c c	✗
2-6	c a c c b	✗
3-7	a c c b a	✓
4-8	c c b a b	✗
5-9	c b a b c	✗
6-10	b a b c a	✗
7-11	a b c a c	✓

Ans = 3

1) Sliding Window + 2 HM

B: a b a c c

HM: { a: 2, b: 1, c: 2 }

A: a b c a c c b a b c a c

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

	Add	Remove		HM
0 - 4			✓	{a:2, b:1, c:2}
1 - 5	5	0	X	{a:1, b:1, c:2}
2 - 6	6	1	X	{a:1, b:1, c:3}
3 - 7	7	2	✓	{a:2, b:1, c:2}
4 - 8	8	3	X	{a:1, b:2, c:2}
5 - 9	9	4	X	{a:1, b:2, c:2}

O(1)

H.W: Try the code for this problem.

T.C =

initial value of e = M-1 s

final = N-1 e

(N-1) - (M-1) + 1

N-M+1 × O(26) + M + M

N >> M

O(N)

Q Given a string of length N.

Return the sum of ASCII values of every substring of length K (Return array)

a → 1
b → 2
⋮
z → 26

a c b a d e b c e K=3

6 6

Sliding Window

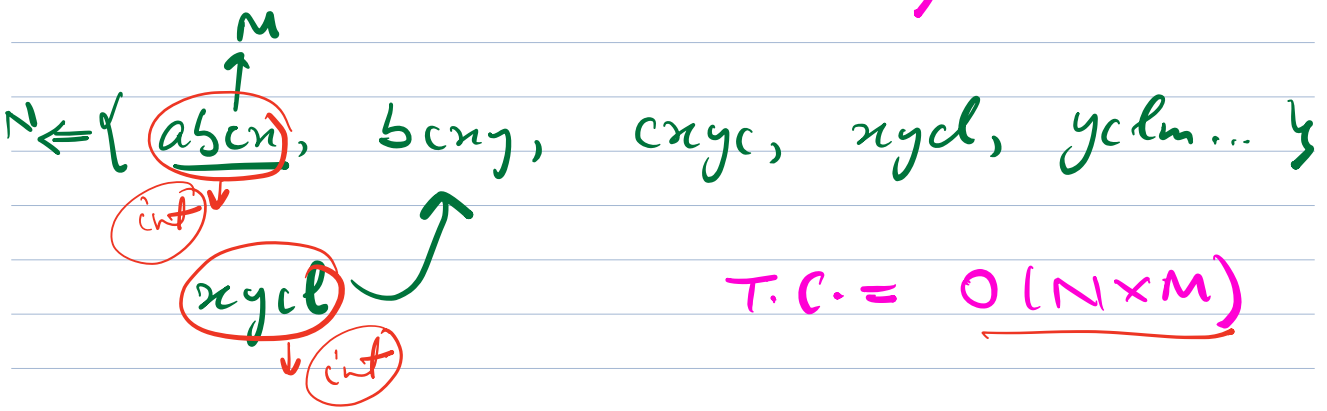
1. Window size should be fixed
2. Ans of next window can be found by removing & adding something to the current answer.

Q Find a given pattern in a large text

(N)
Text: a b c x y c l m o x y c l j p q x y c m k r t x y c l
Pat: x y c l
(M)

Brute force

T.C. = O(N×M)



T.C. = O(N×M)

N { 71, 81, 50, 69, 59, 12, 7.. }

12

O(N)

Comp of 2 similar string = O(length)

" " 2 integers = O(1)

1) Sum of ASCII values

a b c → 97 + 98 + 99 = ___

(N)

Text: abcxycl mo xycl ; p 7 x yc m k r t xycl

a

Pat: x y c l
(M)

Sum of ASCII values

ag → 97 + 99

bb → 98 + 98

1) If sum of ASCII values of pat != sum of ASCII values of substr
⇒ Definite mismatch

2) If the values match.
then compare char by char

Best Case ⇒ T.C. = O(N)

Worst Case ⇒ T.C. = O(N×M)

Collision

h(aa5c) = h(aaad)

[0-9]

1 2 3 = 1+2+3

$321 = 3+2+1$ ✗

$231 = 2+3+1$

$123 = 1\times10^2 + 2\times10^1 + 3\times10^0$

$321 = 3\times10^2 + 2\times10^1 + 1\times10^0$

$231 = 2\times10^2 + 3\times10^1 + 1\times10^0$

26

a b c $\Rightarrow$ $a+b+c$ ✗

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

a → ASCII

b c a

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

c a b

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

$p \geq 27$

$a = 0$

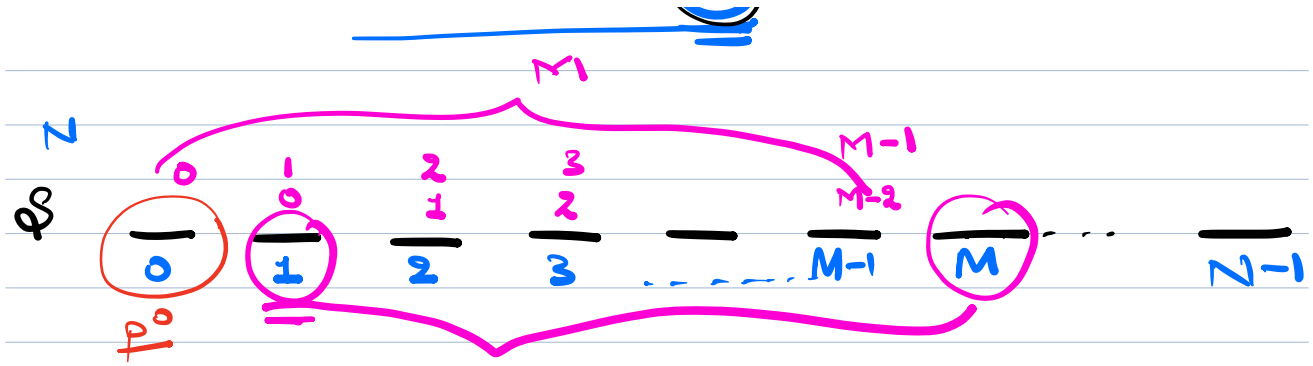
$aaa = 0$

$aa = 0$

[a – z]

97 98 ....

1 2 26



$$h = S[0] \times p^0 + S[1] \times p^1 \cdots\cdots S[M-1] \times p^{M-1}$$

$O(1)$

$$h_{next} = S[1] \times p^0 + S[2] \times p^1 \cdots\cdots S[M-1] \times p^{M-2} + S[M] \times p^{M-1}$$

$$h_{next} = \left(\frac{h - (\text{ASCII of char we are removing})}{p} \right) + (\text{ASCII of char we are adding}) \times p^{M-1}$$

Overflow

$\therefore K = 10^9 + 7$

Collision

Worst Case = O(N x M)

H.W. Try to implement Rabin Karp String Matching Algo

Rolling Hash function