



# Mega Class: Strings

Special class

① LC 1781: Sum of Beauty of All  
Substrings

2 Min to Read On

Beauty of  $s \Rightarrow (\text{Most freq} - \text{Least freq})$   
What to do?  $\rightarrow$  all substring's Beauty Sum

$S = 'aabcb'$

$a \rightarrow 0$

$aa \Rightarrow 0$

$aab \Rightarrow 1$

$aabc \Rightarrow 1$

$aabb \Rightarrow 1$

$a \Rightarrow 0$

$ab \Rightarrow 0$

$abc \Rightarrow 0$

$abcb \Rightarrow 1$

$b = 0$

$b \leftarrow 0$

$bcb \Rightarrow 1$

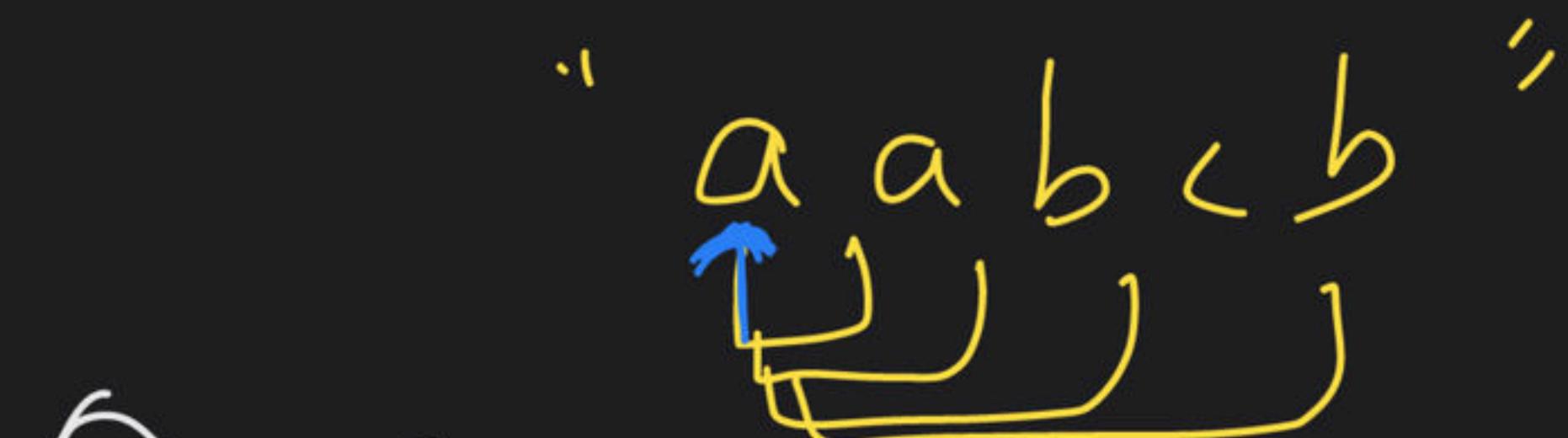
$c = 0$

$cb \Rightarrow 0$

$b = 0$

$\Rightarrow$

$\sum_{i=1}^n -\lceil \frac{c_i}{b} \rceil$

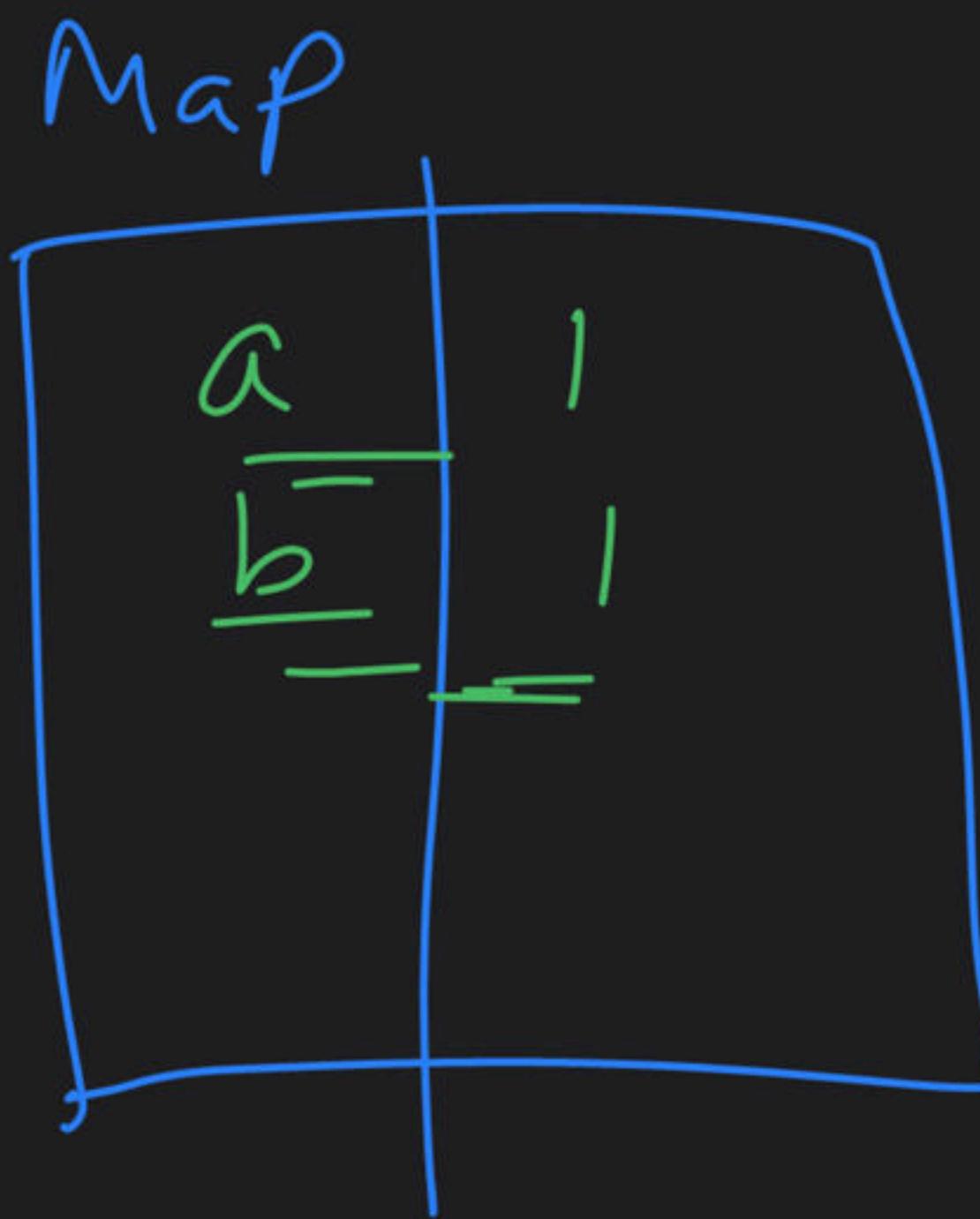
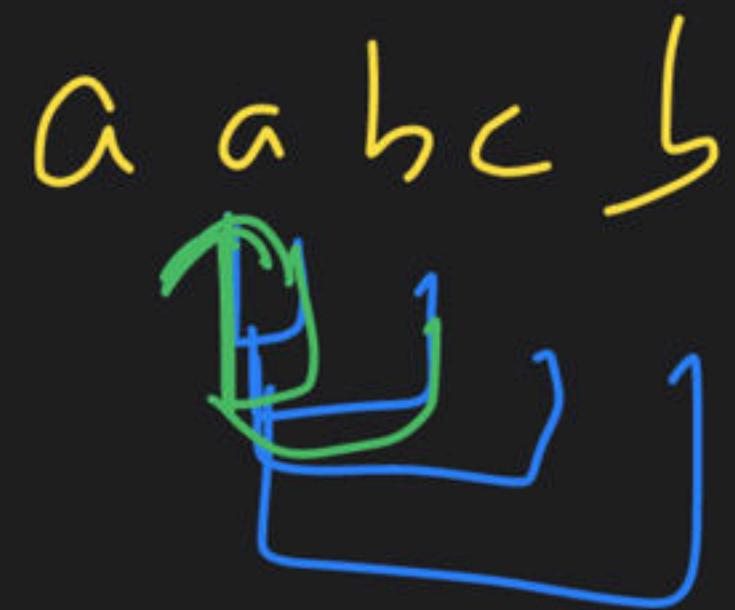
"aabcb" :  

  
 ① find all substrings [nested loop]

- ① a  $\Rightarrow 1 - 1 = 0$
- ② a a  $\Rightarrow 2 - 2 = 0$
- ③ a a b  $\Rightarrow 2 - 1 = 1$
- ④ a a b c  $\Rightarrow 2 - 1 = 1$
- ⑤ a a b c b  $\Rightarrow 2 - 1 = 1$

Map

a	x2
b	x2
c	v

$\alpha \rightarrow$   
 $\alpha b \Rightarrow$



$a b c a b c b$

$\left\{ \begin{array}{l} a \\ ab \\ abc \\ abca \\ cbcab \\ abcabc \\ abcabcb \end{array} \right.$

$\longrightarrow$

$\longrightarrow$

$\longrightarrow$

$\longrightarrow$

$\longrightarrow$

$\longrightarrow$

$\longrightarrow$

$a - x_2$   
 $b \rightarrow x_2 x_3$   
 $c = x_2$

$a b c a b c b$



a b c    a b c b

a  
a b  
a b c  
a b c a

$\bar{a} = \text{Fog}$   
 $b = 1$   
 $c = 0$

## 2 Sherlock & Anagrams (hackerrank)

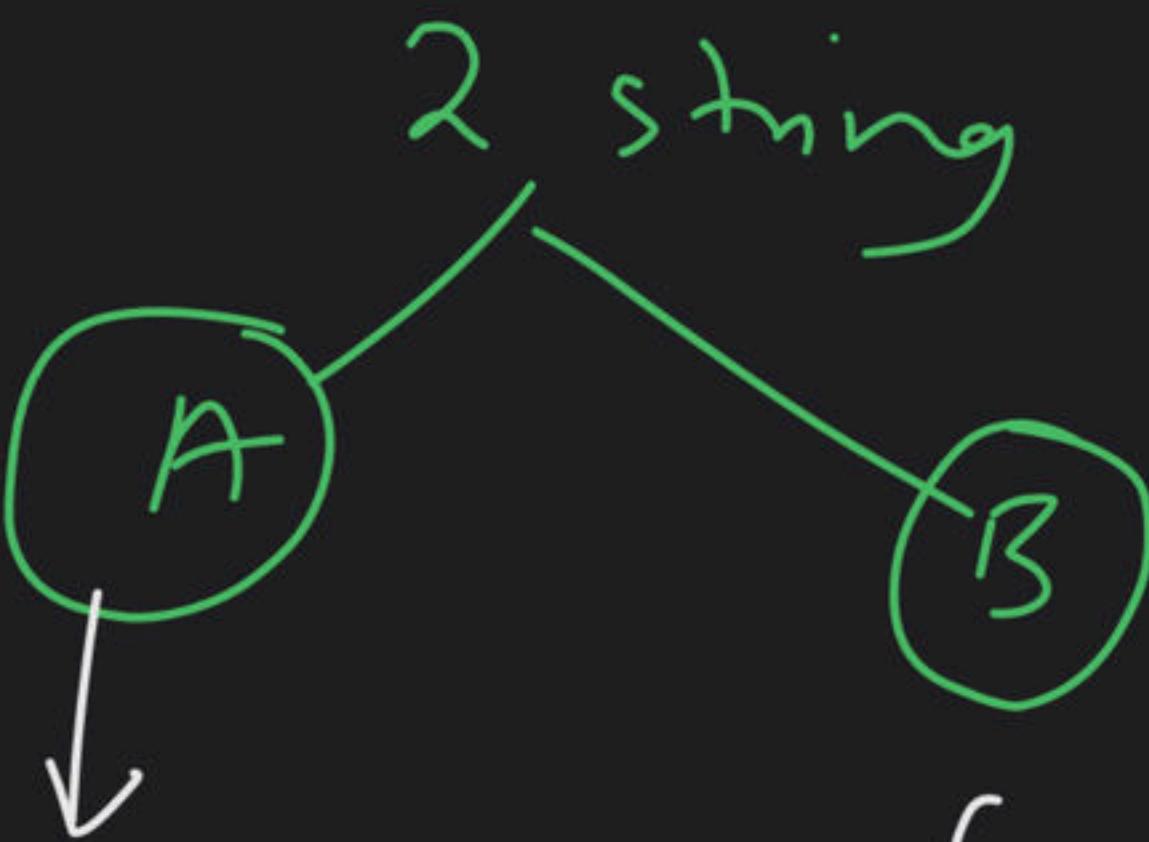
mom



?

Anagrams

mmo



freq. Map

$$m = 2$$

$$\theta = 1$$

freq. Map

$$m = 2$$

$$\theta = 1$$

m o m

=

Sort ✓

m m d

Sort

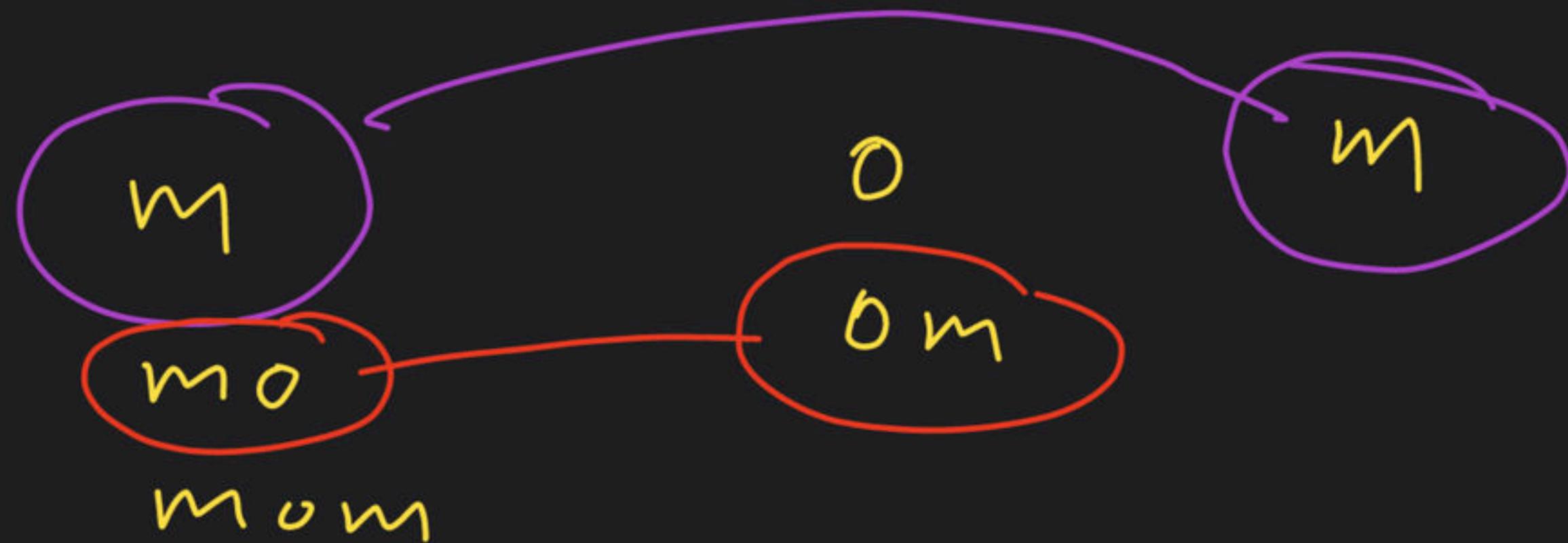
m m d

equal

m m d

Same => ✓ Anagram  
=

$m_0 m$

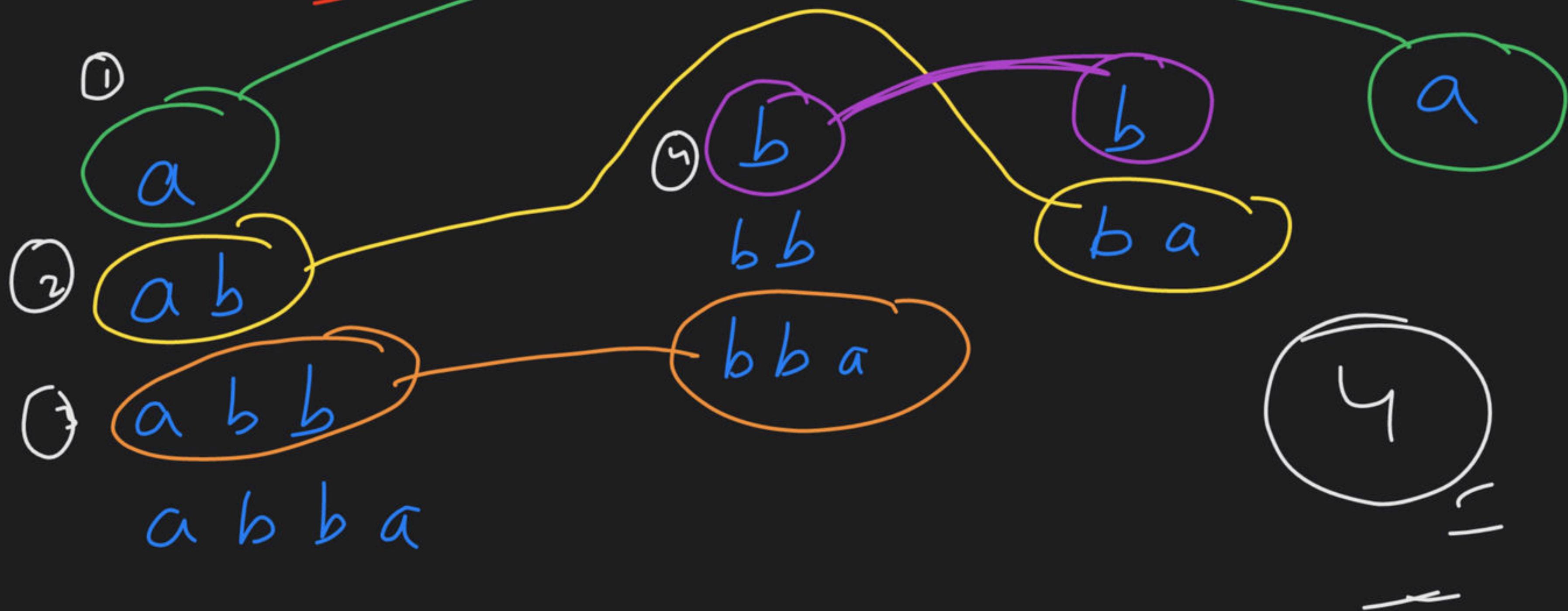


Total

(2)

abba

=



a b b a

ab - ba

✓  
a  
ab

ab b ✓

b  
bb

bb a

b  
ba

a b . b a

✓  
nested

O(<sup>no. of sub</sup>)

=

B  
S.O. →

dr<sup>2</sup> → ①

extra space → ②

a

Code  
H.W.

final all  
substrings

store  
10

Loop over stored substr  
to check anagram or not.  
start →

abb

a**b**b    bba    bab

$\Rightarrow$  Cal. no. of anagrammatic pairs

$\Rightarrow$  Count > 1

$$\frac{N(n)}{2}$$

(1)

$$a = 2 \Rightarrow \frac{2(2-1)}{2} \Rightarrow 1$$

(2)

$$ab \Rightarrow 2 \Rightarrow \frac{2(2-1)}{2} = 1$$

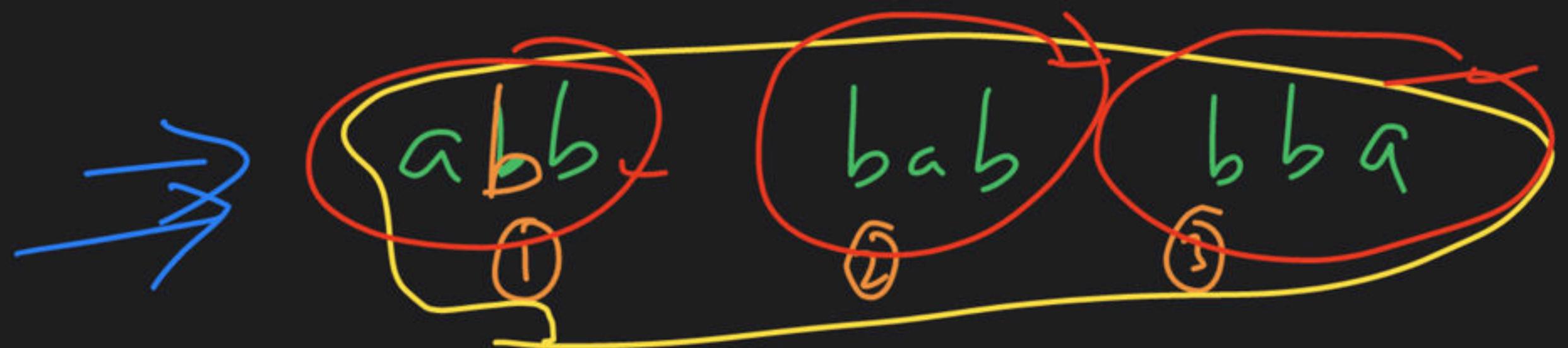
(3)

$$abb \Rightarrow 2 \Rightarrow 1$$

$$b \Rightarrow 2 \Rightarrow 1$$

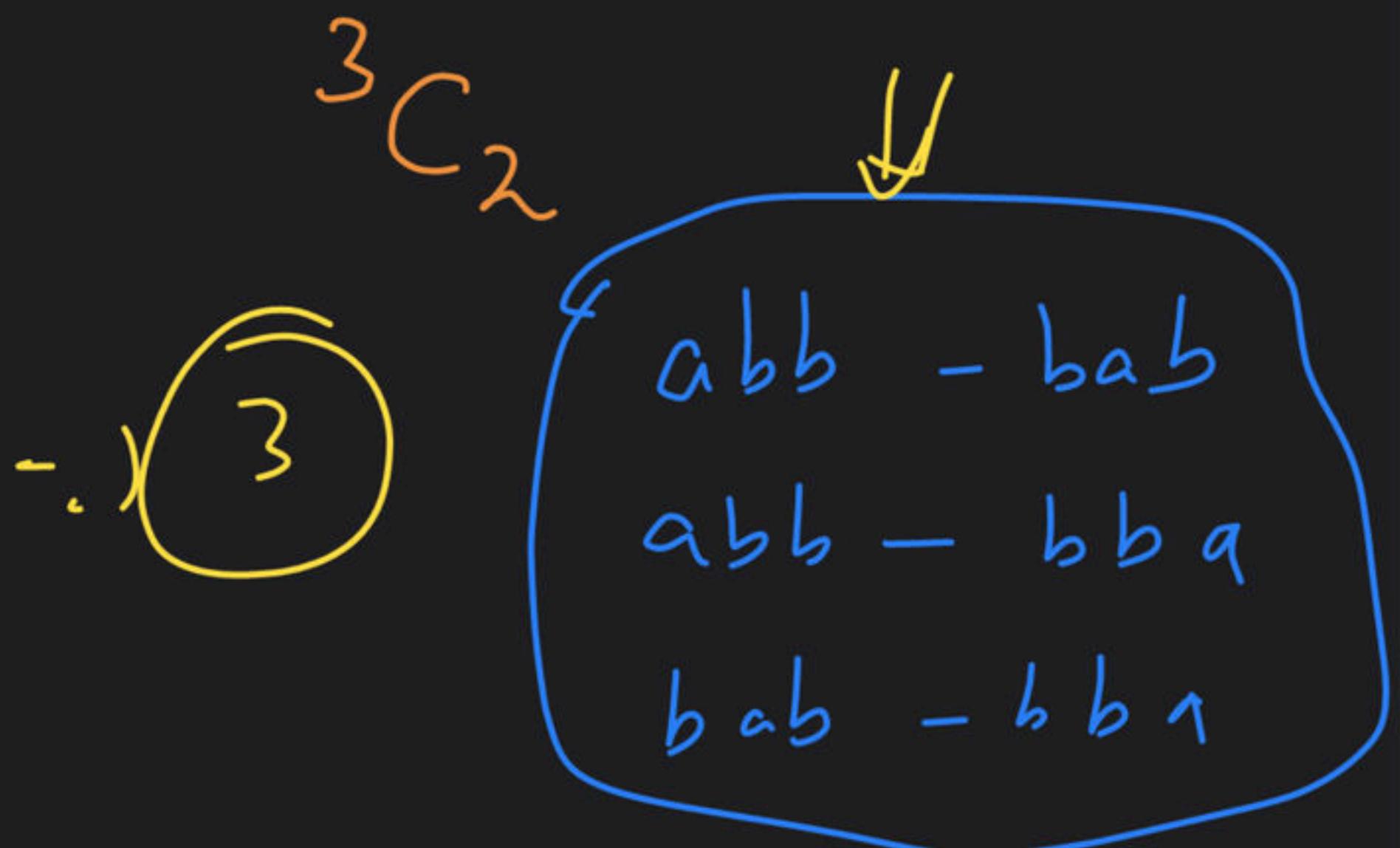
Map substrings - counts

String	int
a	x 2
ab	x 2
<u>abb</u>	x 2
aabb	1
b	x 2
bb	1



$$\Rightarrow {}^3C_2 \Rightarrow \frac{3(3-1)}{2}$$

$$\Rightarrow \cancel{\frac{3(2)}{2}}$$



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

$$N C_2$$

$$N C_2 =$$

$$\frac{N!}{(n-2)! \cdot 2!}$$

$\Rightarrow$

$$\frac{N(n-1)(n-2)!}{(n-2)! \cdot 2!}$$

$$N C_2$$

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

- ① generate all Possible substrings.
- ② Sort each substring
- ③ Count occurrence of each sorted string  
→ Map
- ④ Cal. pairs ↳ ✓ ✓  


C d c d

C  
C d

C d c  $\rightarrow$  C c d

C d c d  $\rightarrow$  C c d d

d  
d c  $\rightarrow$  c d  
d c d  $\rightarrow$  c d d

C c d

sh	int
----	-----

c  $\times 2$

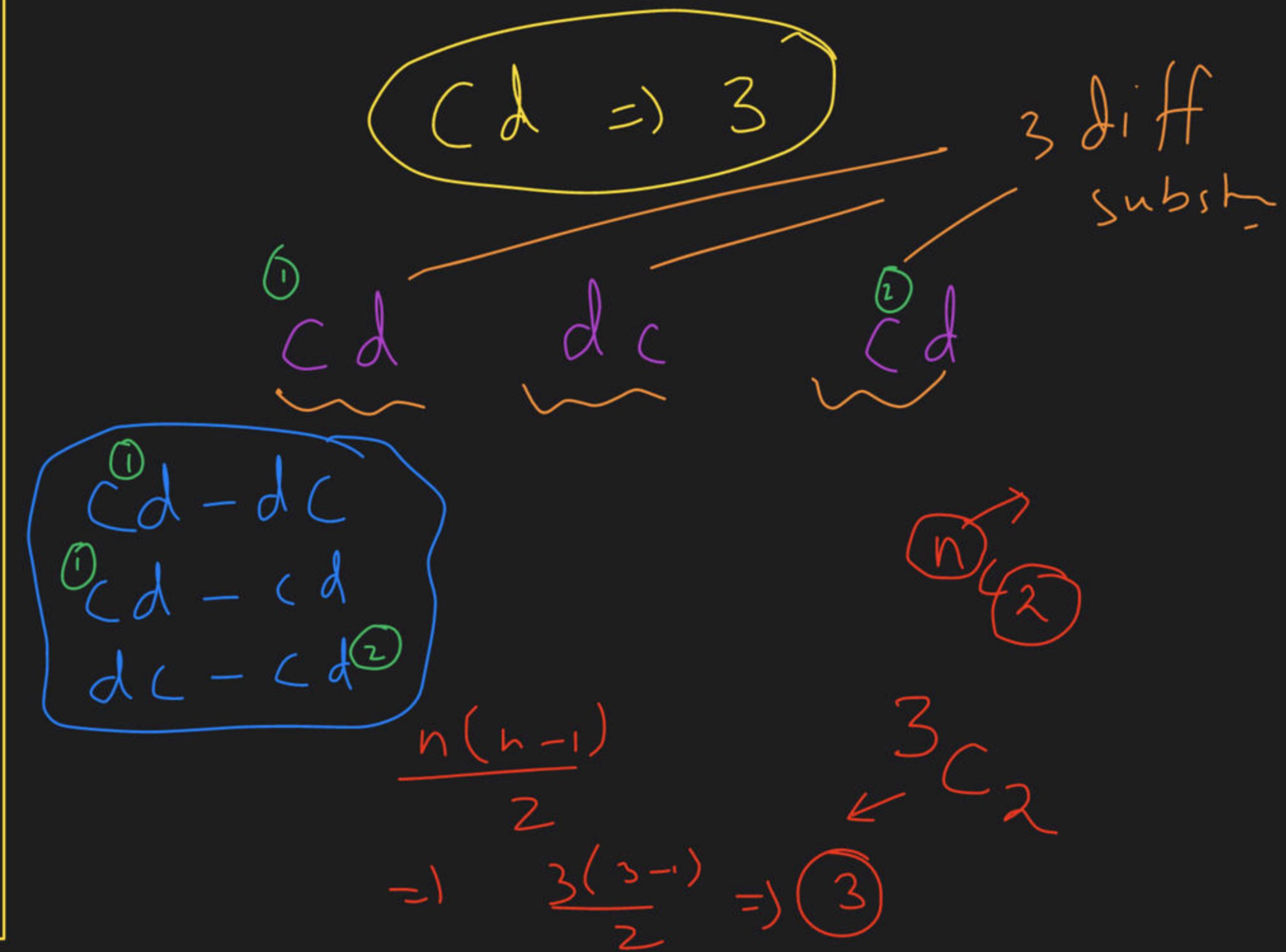
C d  $\times \cancel{x} 3$

C c d |  
C c d d |

d  $\times 2$

C d d |

$C$	2
$Cd$	3
$d$	2



①

Chen. substr

a b c d  
0 1 2 3  
i ↑  
j ↓

outer loop  $\rightarrow i = 0 \rightarrow i \leq 3$

inner loop  $\rightarrow j = 1 \rightarrow j \leq 3$

s.substr(i, j-i+1)  $\rightarrow (0, 4)$   
 $(0, 3)$

③  $\text{LC} \Rightarrow |\Sigma|$ : Reverse words in a string.

$s \text{ min } Break;$   $\Rightarrow$  Read the  
Question

==

"hello world"
   
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
   
 $l = 0, \text{ start} = 0, \text{ end} = 16$ 
individual char. reverse
 $\Theta(1)$ 
  
= In-place

=>
   
 "d l y o w - - o l l e h - - "
   
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
   
→ after reversing whole string
  
Only 2 things left
  
① Remove extra space
② individual word reverse

- - d l y o w - - o l l e h - - -  
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
 i

$l = 0$

$\underline{\text{start}} = 0$   
 $\underline{\text{end}} = 0$

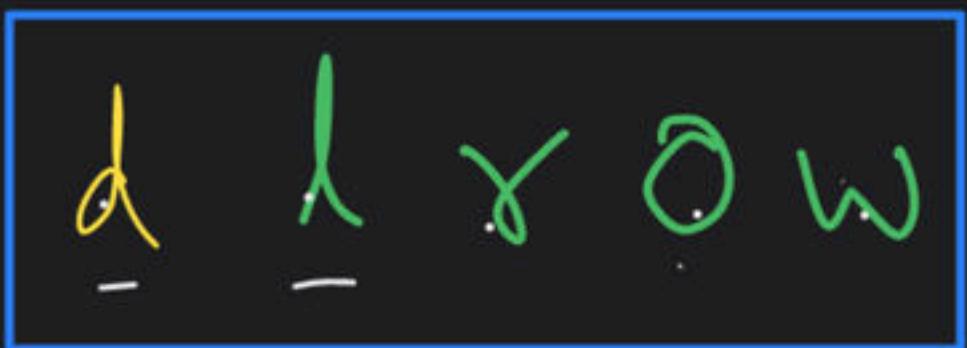
individual  
word  
year

- ① skip leading space

while ( $s[i] == ' '$ )  $\checkmark$   
 i++  
~~i++~~

- ② copy char. of a word to the correct position
- $i < n$
- $\leftarrow$
- while ( $s[i] != ' '$ )  
 {  
 $s[\underline{\text{end}} + 1] = s[i + 1];$

" - - d l r o w - - o l l e h - - - "

"  low - - o l l e h - - -  
0 1 2 3 4 e  
→

③ Reverse individual word

reverse( s.begin() + start , s.begin() + end );

w o r l d   o w - - o l l e h - - -  
e

⑨ add space b/w words  
= s[end] = " "

end++

w o r l d - w - - o l l e h - - -

↑      ↓  
s      i  
e

start = end;

we mark one iteration completed

complete =

++

w o r l d - w - - o || e h - - -

s,e

w o r l d - o l l e h | e h - - -

rewer  
=

0 1 2 3 4 5 6 7 8 9 10 11 12 13  
w o r l d - h e l l o - e h



l

m

15

16

n

e,s

e-i

$s.\text{resize}(n)$

$\equiv$



no. of char

final step.  $s.\text{resize}(\underline{\text{end}-1})$

$\hookrightarrow$

a - good - - - example

$\Rightarrow$

"  
e | p m a n e  
|



i

↑

e

s

l  
- - d o o g - a '

$s, e, l = 0$

example -

l  
d v o g - a

e

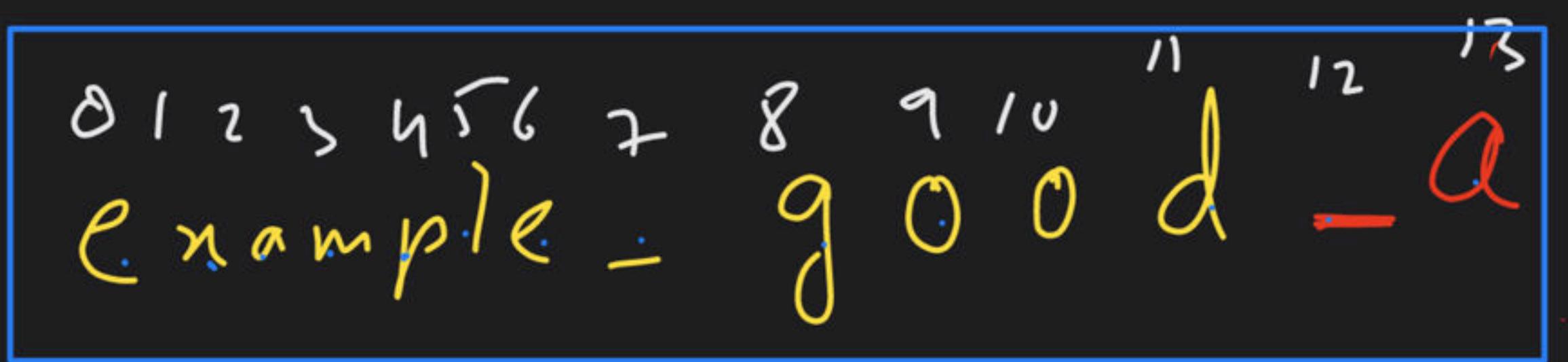
s

example - d o ~~o~~ ~~g~~ ~~g~~ - a

exercice

s

Example -  dog - a

Example -  good - a

i

e

15 l

g

e  
s

e-i

$\Rightarrow$

$2 \min$   
 $=$

Then next  $Q_m$   
 $=$

$\Rightarrow$

"ie "abcd ef gh ij k -"  
e  
i  
 $O(n)$

Lc<sub>i</sub> 68  
= 2 min Break  
=

Tent Justification

↓  
S min

HARD



To read  
Question  
=

①

Spaces added b/w words

→ at least one space.

) aur spaces Bhi ho skte

hai But man width sp

zada hai overshoot

krne chahiye no. of  
char

This, is, an, example, of text justif.

maxwidth = 16

This - is - an - example

4 1 2 1 2 1 7

=> 18 == 16 ?

X

This - is - an -  
example

leading or trailing  
Spacer hair space  
single word   last

"Justification. - - -"

③ Single word hi aa Para hai line  
me.

⑯

"Acknowledgment. - - "

shall

① last line OR single word in  
line → keep it left  
justified.  
Word me only 1 space

(2)

Other than that full justified.

→ can have

— — — —

> 1 spacer b/w  
word

⇒ No leading OR Trailing  
spacers,

full justified

$\Rightarrow$  When spaces can be evenly divided

$\Rightarrow$  Can't be evenly divided  
 $\hookrightarrow$  greedily left to  
right space add  
=

abc - def = zyz - mn



abc - def - zyz - mn

~~abc~~

abc -- def -- zyz --

①

⑤ → ⑦  
max =

⑧

⑨~~2~~ 6

This is an example of Tent

MW  $\Rightarrow$  16

Justification  
=

$\Rightarrow$  Case 1: Current line is complete.

This is an example  
L U U U U U L  
1 1 2 1 2 1 7

(18)

This is an

{ This is, is, an example } → just the words

This is an example

ut the his speech  
changes

This is , a ~

Y + 1 + , 2 + 1 + 2 + 1 .

11

{ This , is , an } → Current line  
=

$$8 + 3 + 7 = \textcircled{18}$$

$$\textcircled{18} > 16$$

Current word

example.

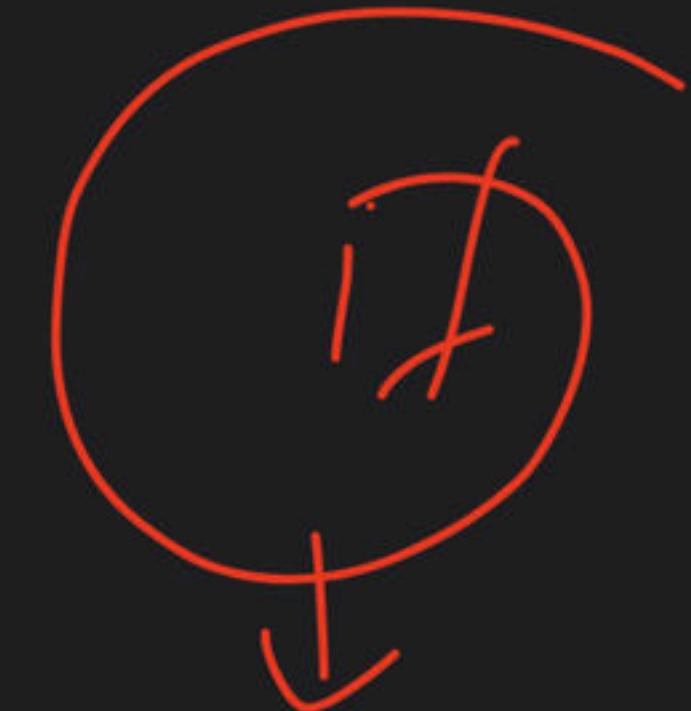
This - is - an - example

Currentline

[This, is, an]

8 + 3 + 7

Currentword = example



I don't need

example

This, is, an ] → ko mila kar  
mujhe poori  
line Banamai hai.

-> ⑧ Manw => 16

Entraspase => Manw - Current line Total, ha =>

$$16 - 8 \Rightarrow ⑧$$

This is an

Fig

6

③

"This is an"

add 8 spaces

=>

8 / (3 - 1)

Spaces in B/w

= extraSpace / (currentLineSize - 1)

=> 8 / (3 - 1) => 8 / 2 = 4

This is an

"This ---- is ---- an"

=> Extraspace => 9

Spaces in b/w  
=> extra space / words - 1

remainder = extra % (word - 1) → 9 / 2 => 4  
→ 9 % 2 => 1

man ~

7

"abc---j"

[This, is, an]

[This---, is---, an]