

Week - 5

char

array



strings

Doubts with Lakshay Bhaiya [Week 6]

Special class

⇒ Binary Search "M N"

Majority Number

=



① Moore's Number

Alg o
= go

② B.S ~~☆☆☆~~

\Rightarrow array
 $=$

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

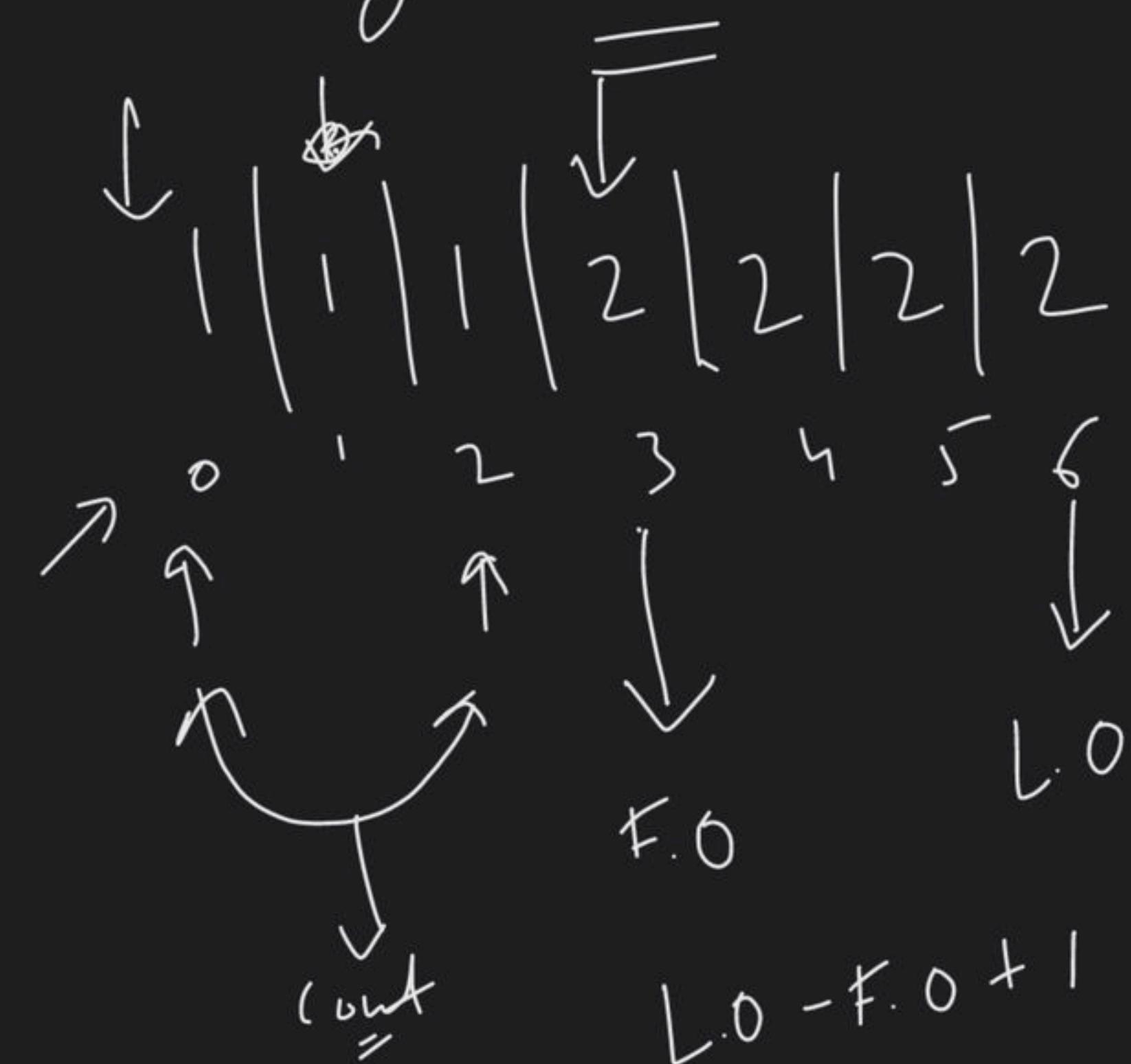
\Rightarrow MI Brute force \Rightarrow Count each element
 $=$

$\exists \rightarrow 0 \rightarrow O(n)$
↓
 $O(n^2)$

M2

Binary Search

⇒



$$\begin{array}{c} y > N/2 \\ \boxed{y > 3} \end{array}$$

$$f.0 \rightarrow 0$$

$$l.0 \rightarrow 2$$

$$l.0 - f.0 + 1$$

$$6 - 3 + 1 \Rightarrow \underline{\underline{4}}$$

Total count =

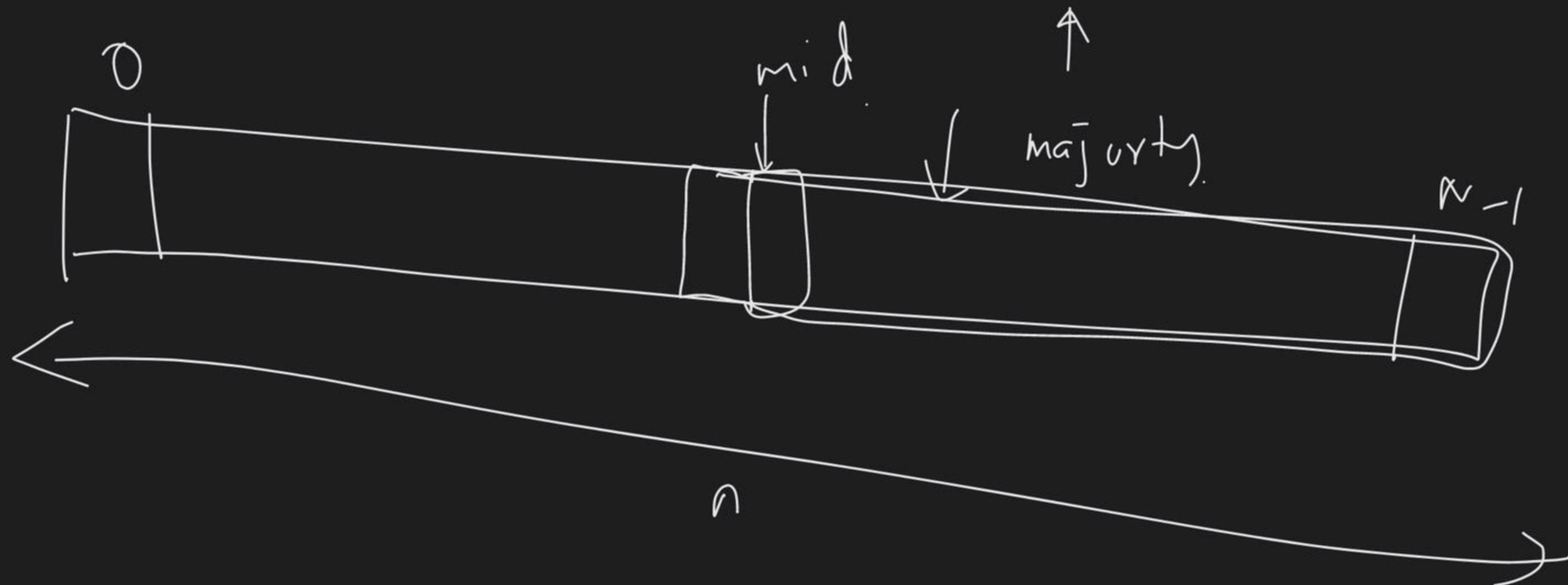
$$l.0 - f.0 + 1 \Rightarrow \underline{\underline{3}}$$

\Rightarrow

$3 | 2 | 3$
0 1 2

\hookrightarrow

$2 | 3 | 3$
0 1 2



\Rightarrow

4 2 2

$$\begin{array}{c}
 2 \left| \begin{array}{c} 3 \\ 1 \end{array} \right| 4 \\
 0 \left| \begin{array}{c} 2 \\ 3 \end{array} \right| 3
 \end{array}
 \quad
 \begin{array}{c}
 2 \left| \begin{array}{c} 5 \\ 4 \end{array} \right| 3 \\
 3 \left| \begin{array}{c} 3 \\ 5 \end{array} \right| 3
 \end{array}
 \quad
 \begin{array}{c}
 3 \left| \begin{array}{c} 3 \\ 7 \end{array} \right| 3
 \end{array}$$

\Rightarrow

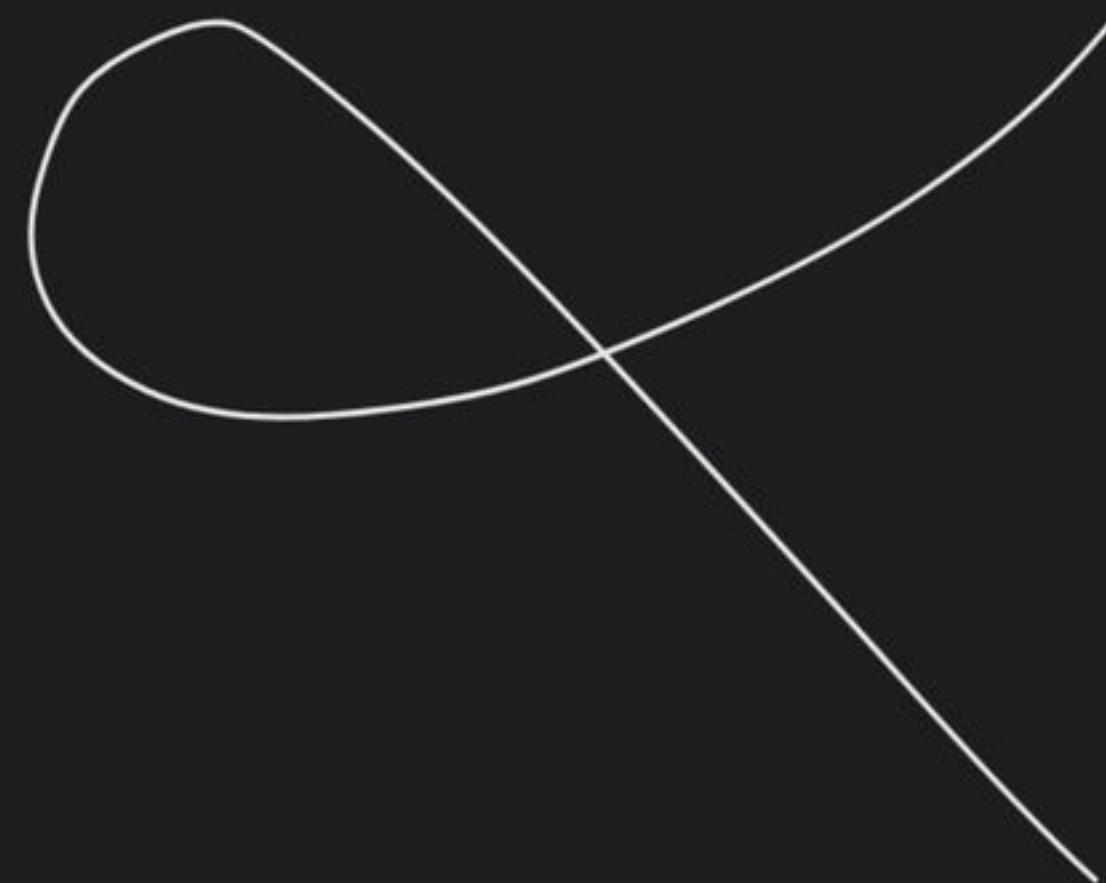
$$\begin{array}{c}
 2 \left| \begin{array}{c} 2 \\ 1 \end{array} \right| 3 \left| \begin{array}{c} 3 \\ 3 \end{array} \right| 3 \left| \begin{array}{c} 3 \\ 4 \end{array} \right| 3 \left| \begin{array}{c} 4 \\ 5 \end{array} \right| 5
 \end{array}$$



$\Rightarrow 3$

z>

1 | 1 | 2 | 2 | 3 | 3) n | y / y
6 1 2 3 4 5 6 7 y



n / 2

l)

8 /
2

4

> 4

① \Rightarrow Majority element always exists
at the mid of the sorted

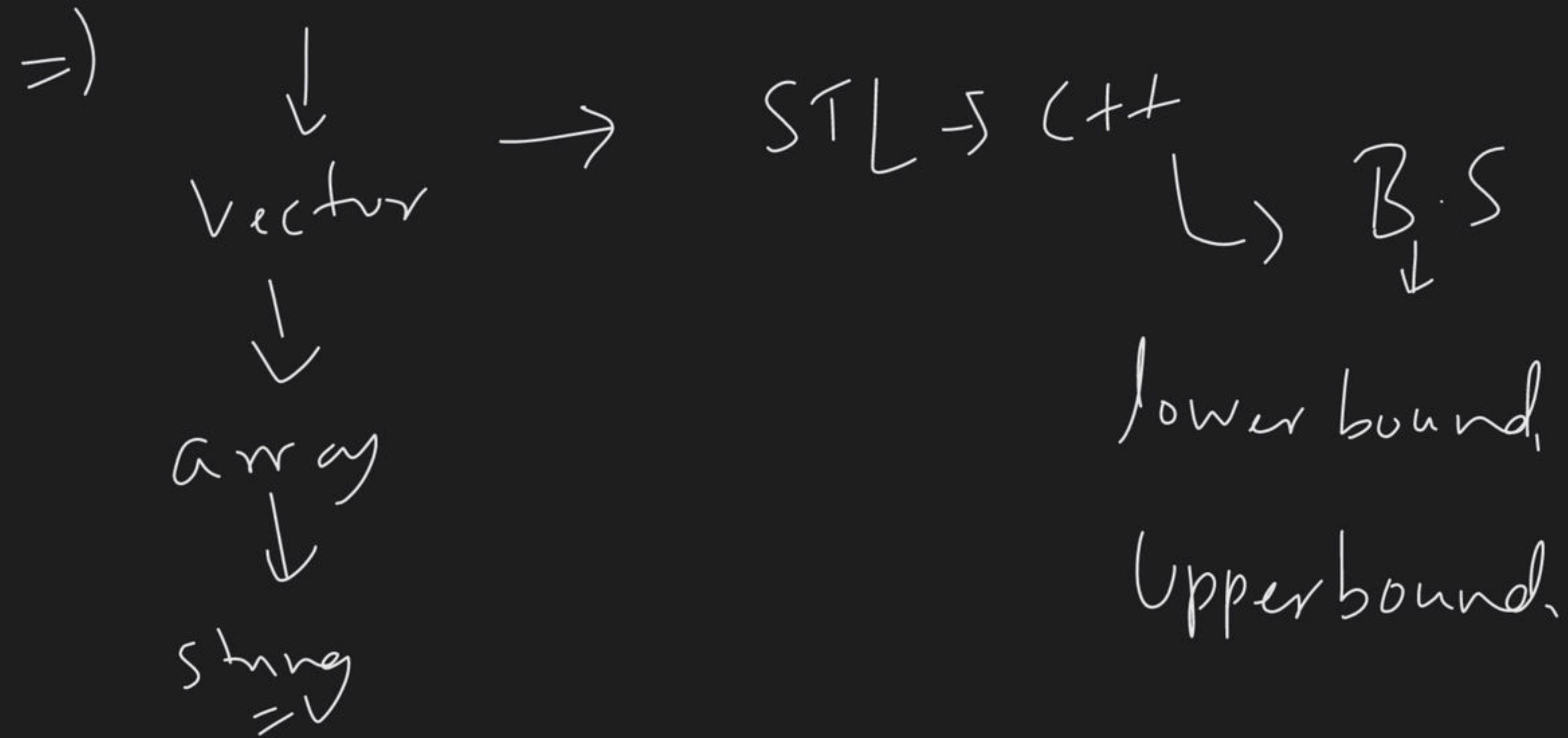
invalid
 \Rightarrow Version of array ✓

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

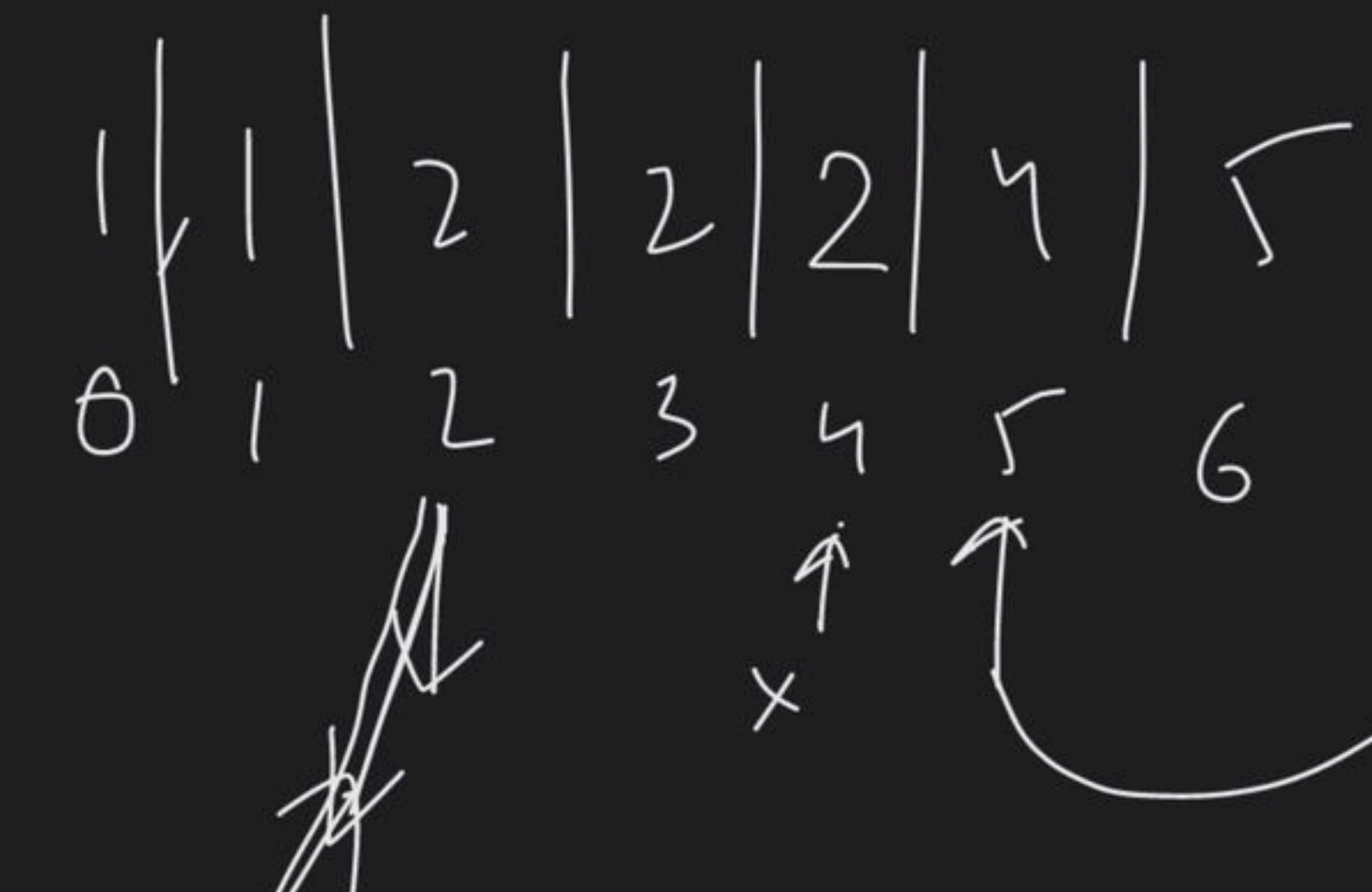
$\Rightarrow T.C \rightarrow O(n \log n)$ ✓

=> Sort() -> C++

=> Binary Search



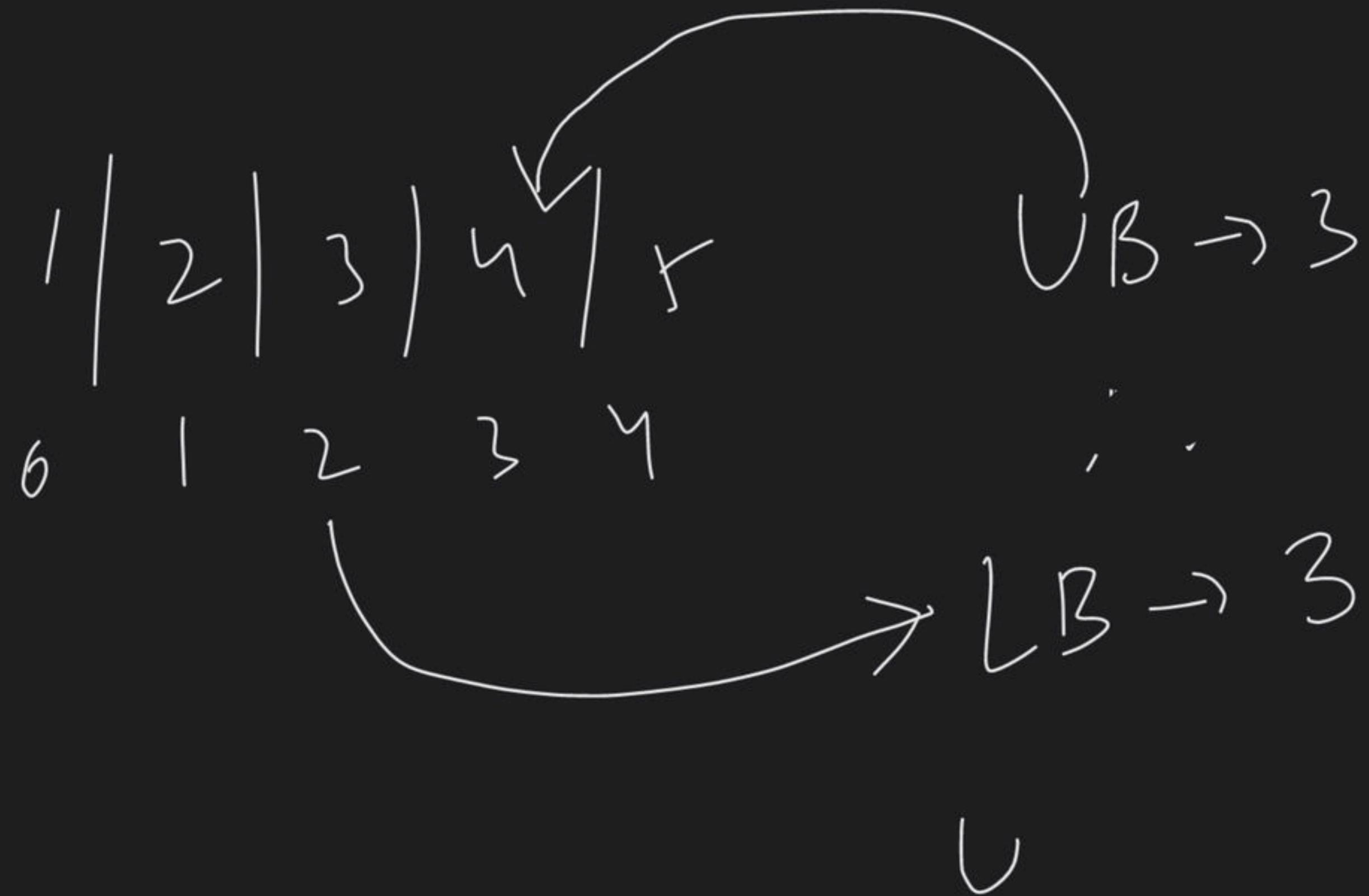
\Rightarrow
Vector
=

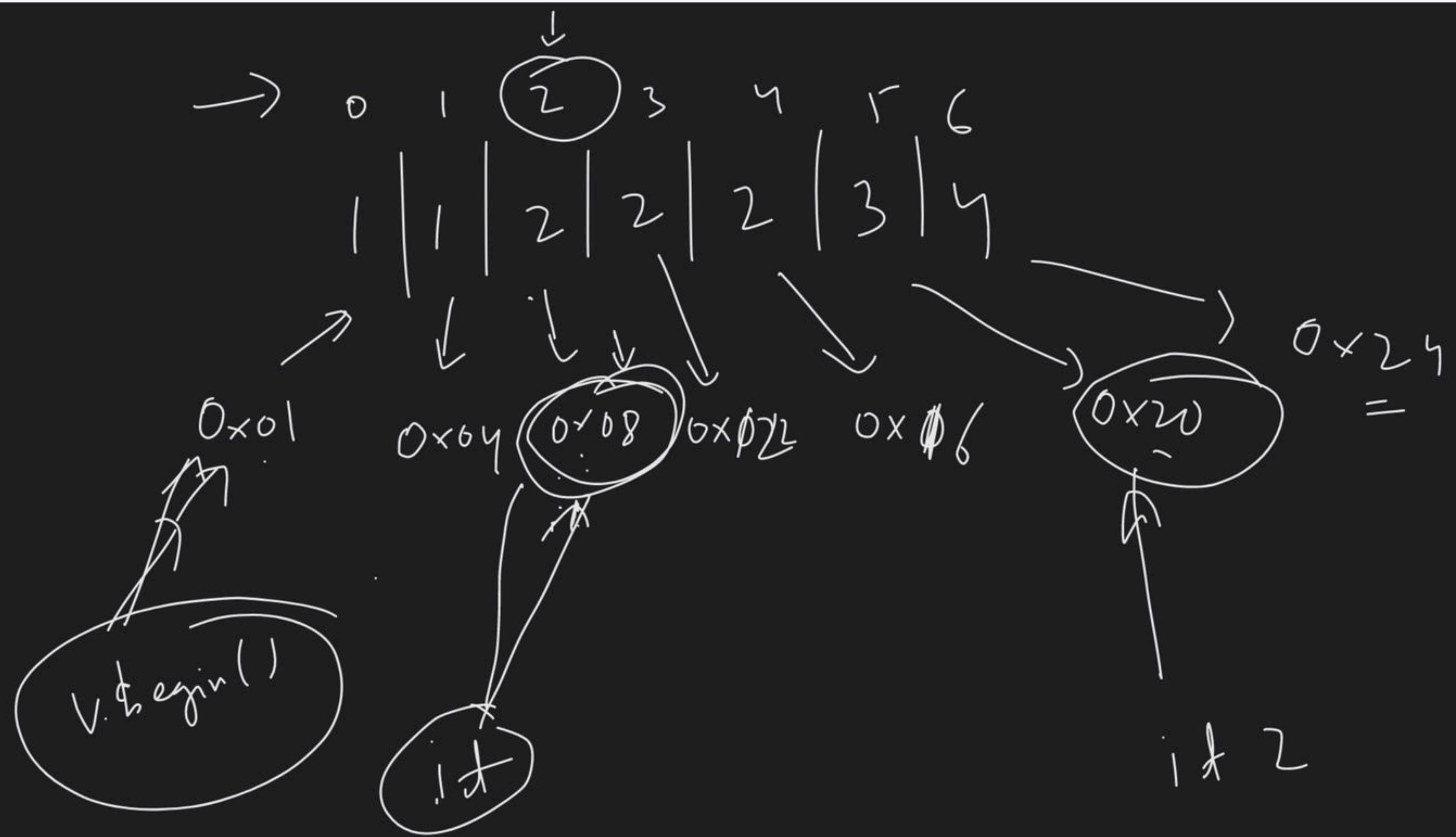


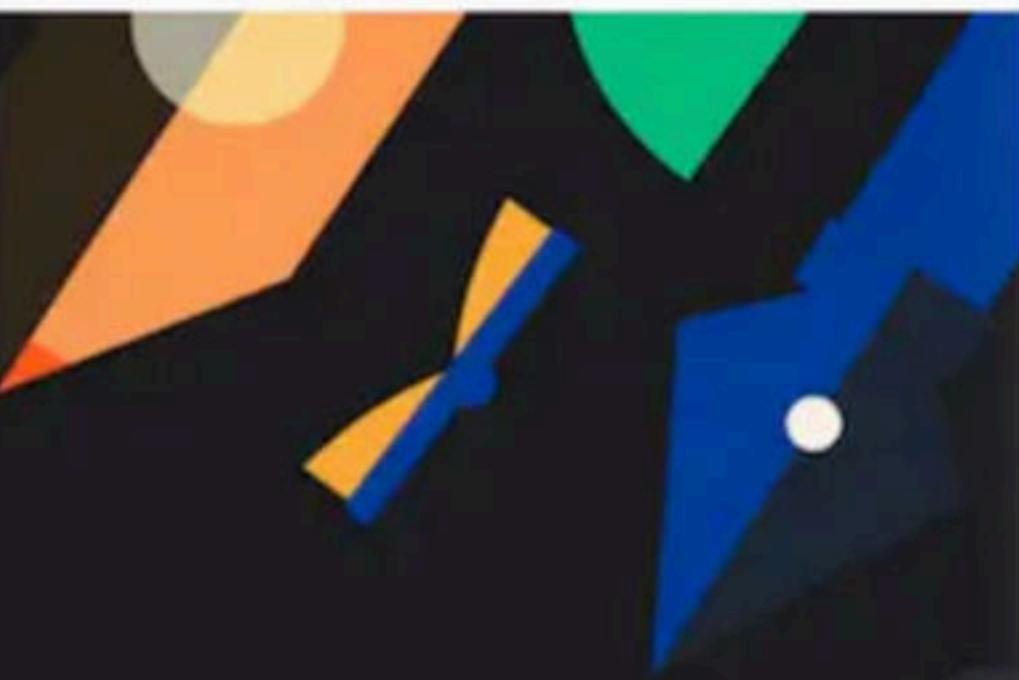
upper_bound(begin,
end,
2);

lower_bound(v.begin(), v.end(), 2)

\ \ 2\ 3\



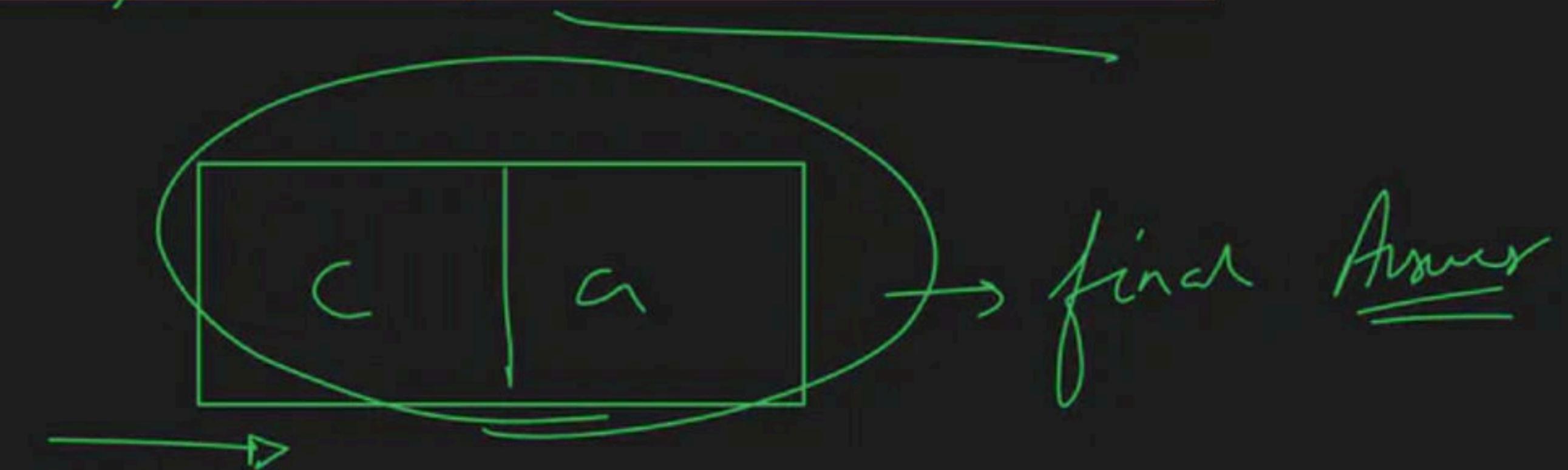
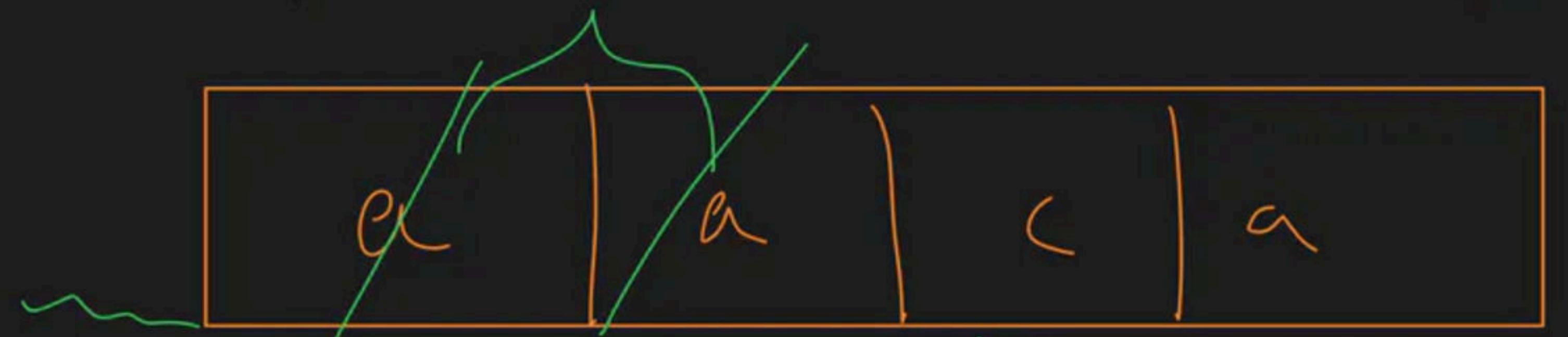
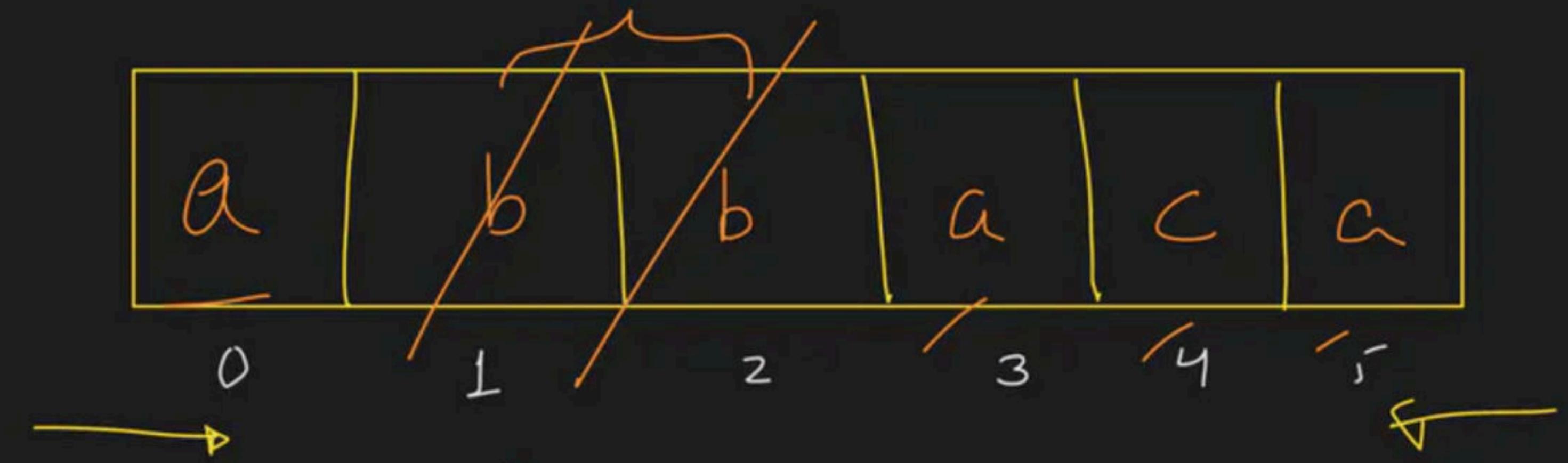


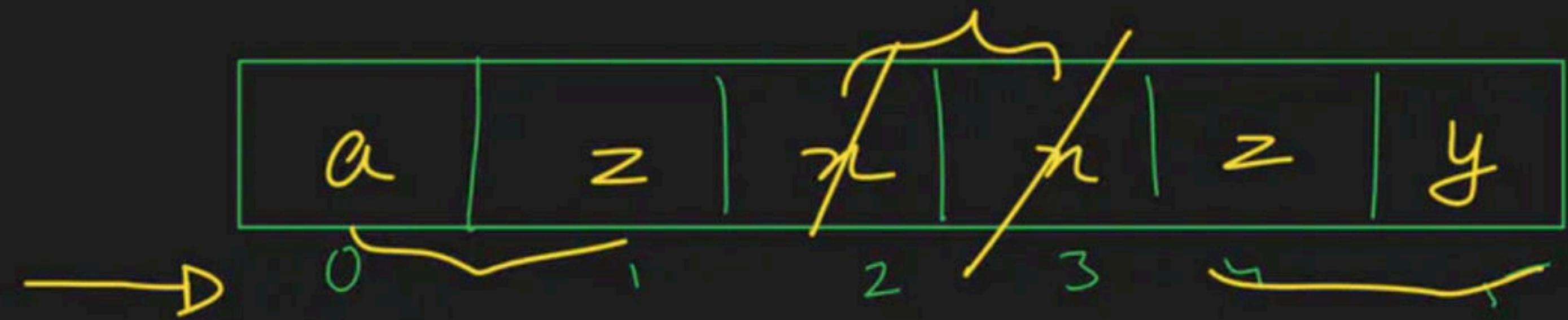


Char Arrays & Strings Class 2

Special class

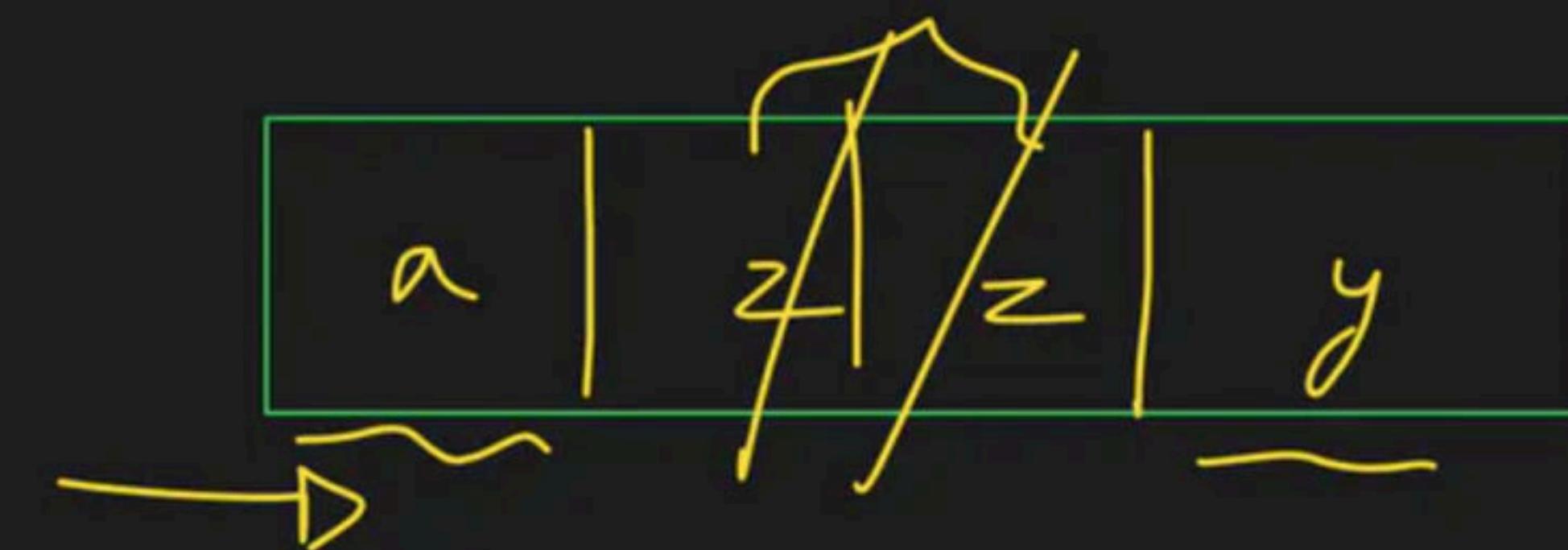




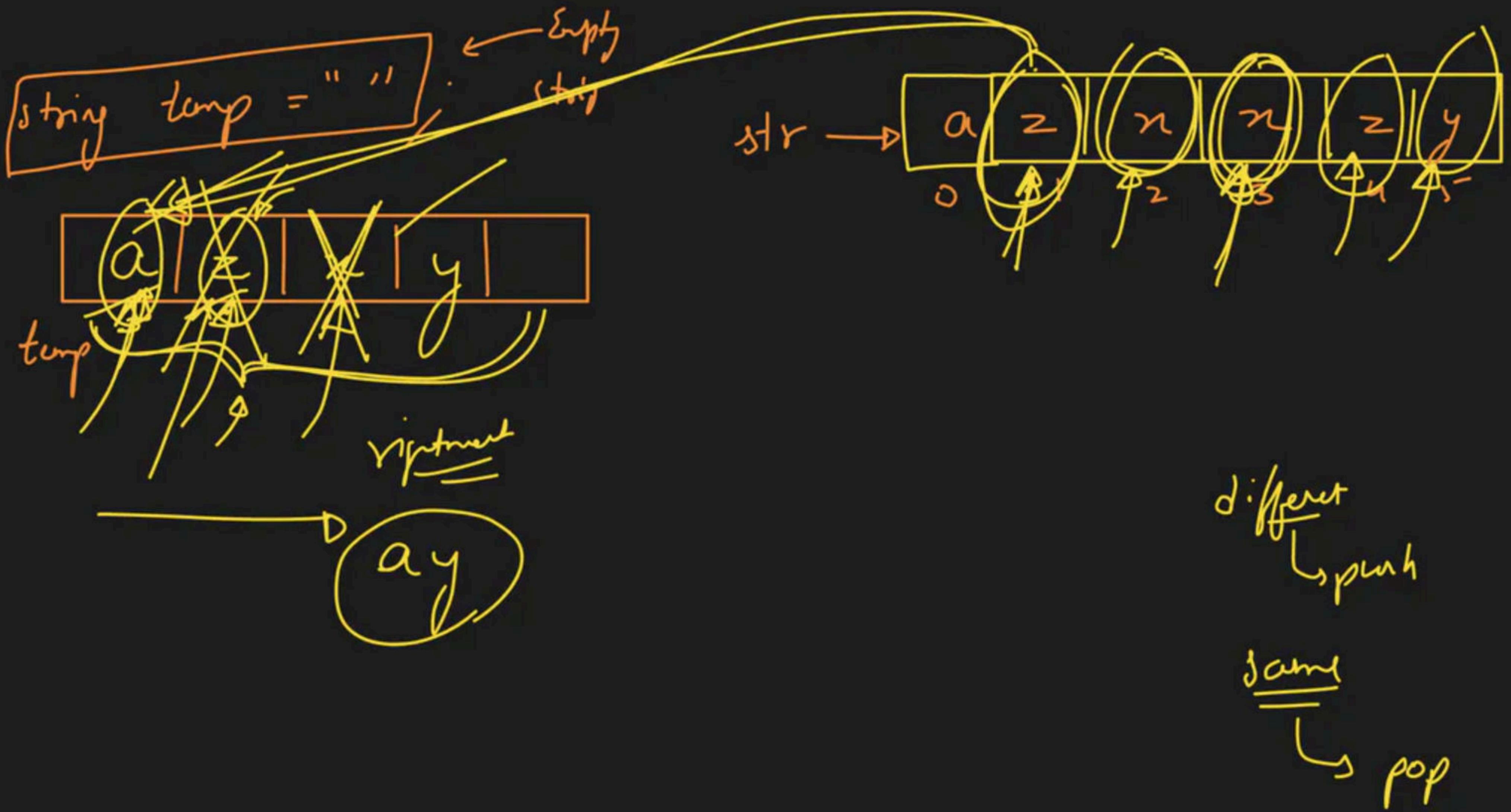


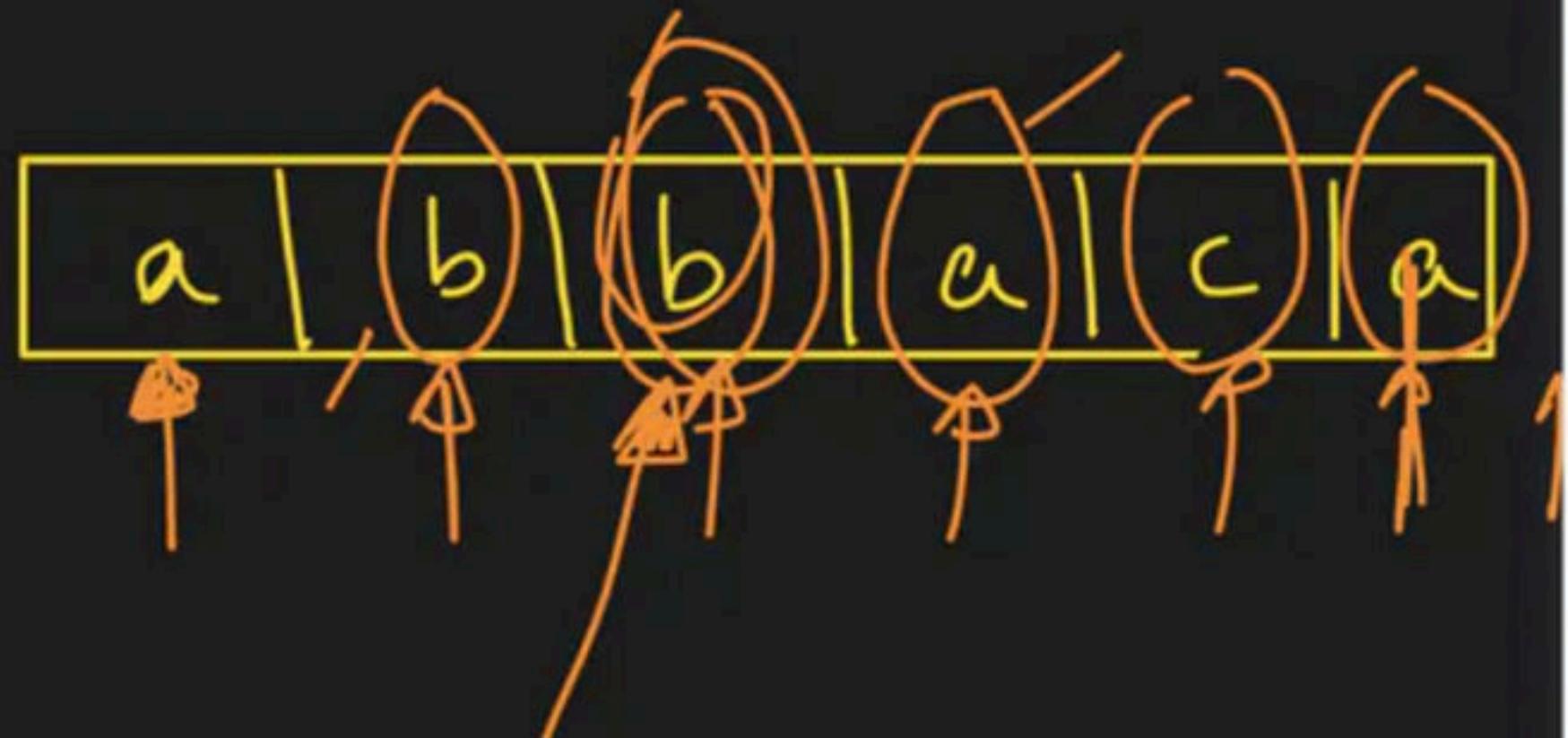
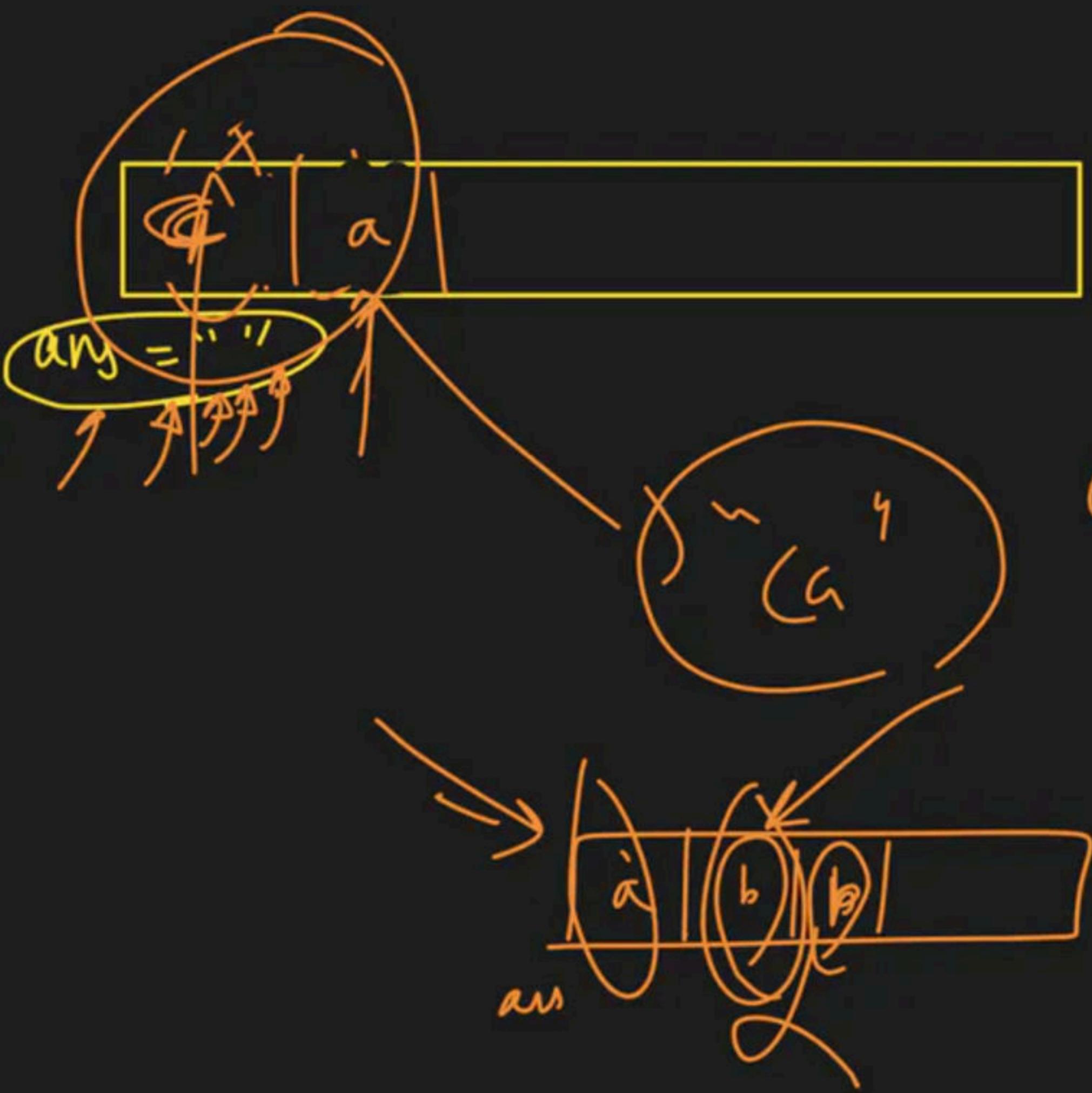
Approach
new string

new empty string



first answer





diff -> push
done -> pop



→ Remove all occurrence of a substring

str →

part →



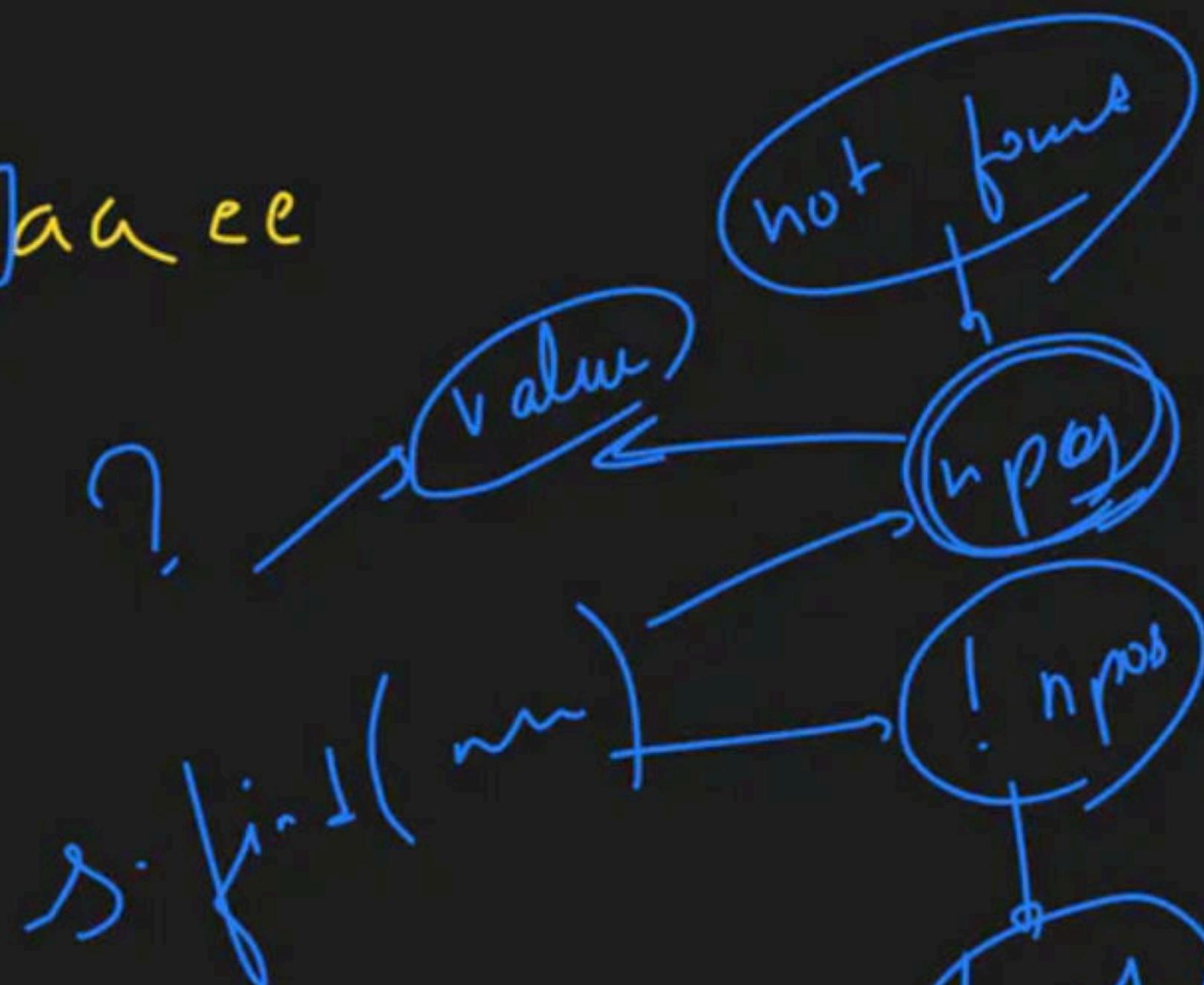
find
ans

while (substring exist in bad string)

remove

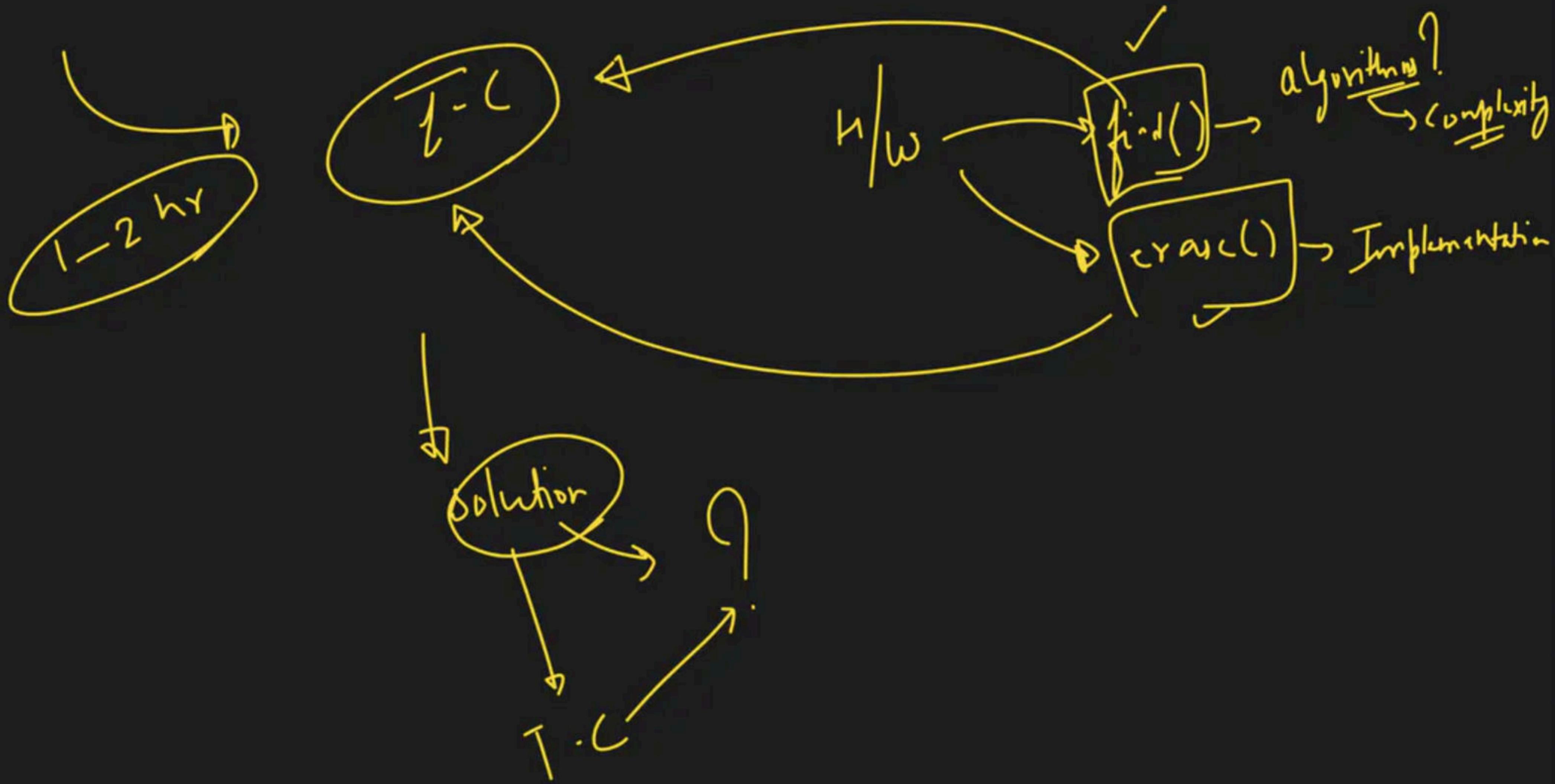
upas

a b c **c** h a b a d d l c b a a c e
a b c b a d d a a c e

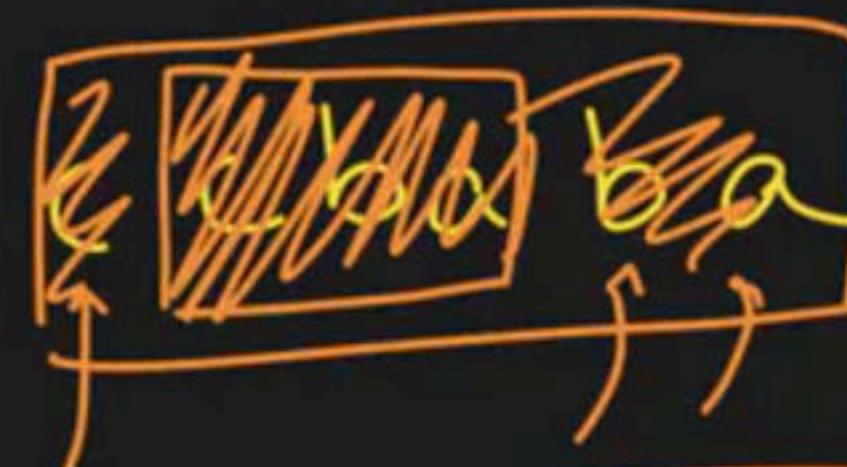


npos

?



aba



dd



part = the

ab added dd → return

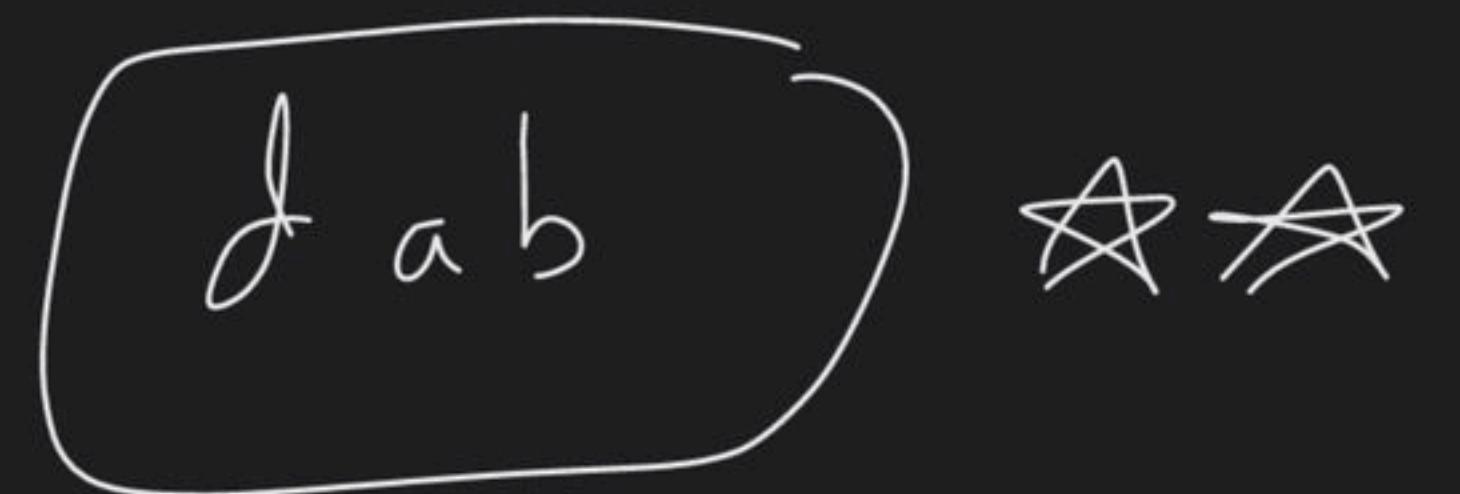
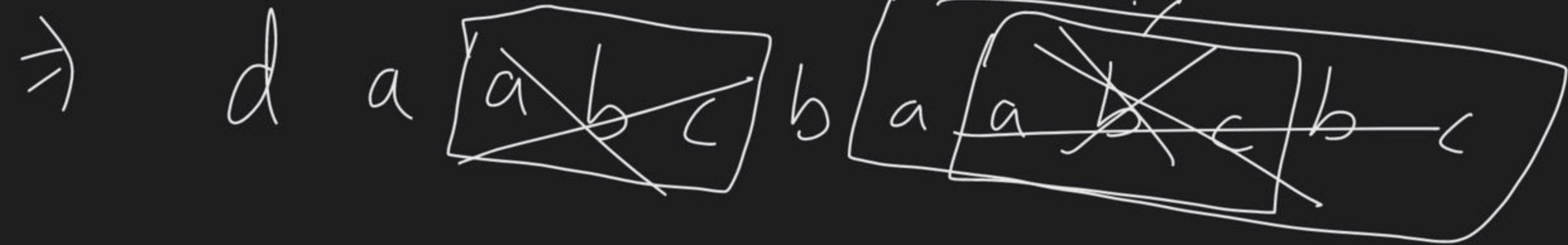
```

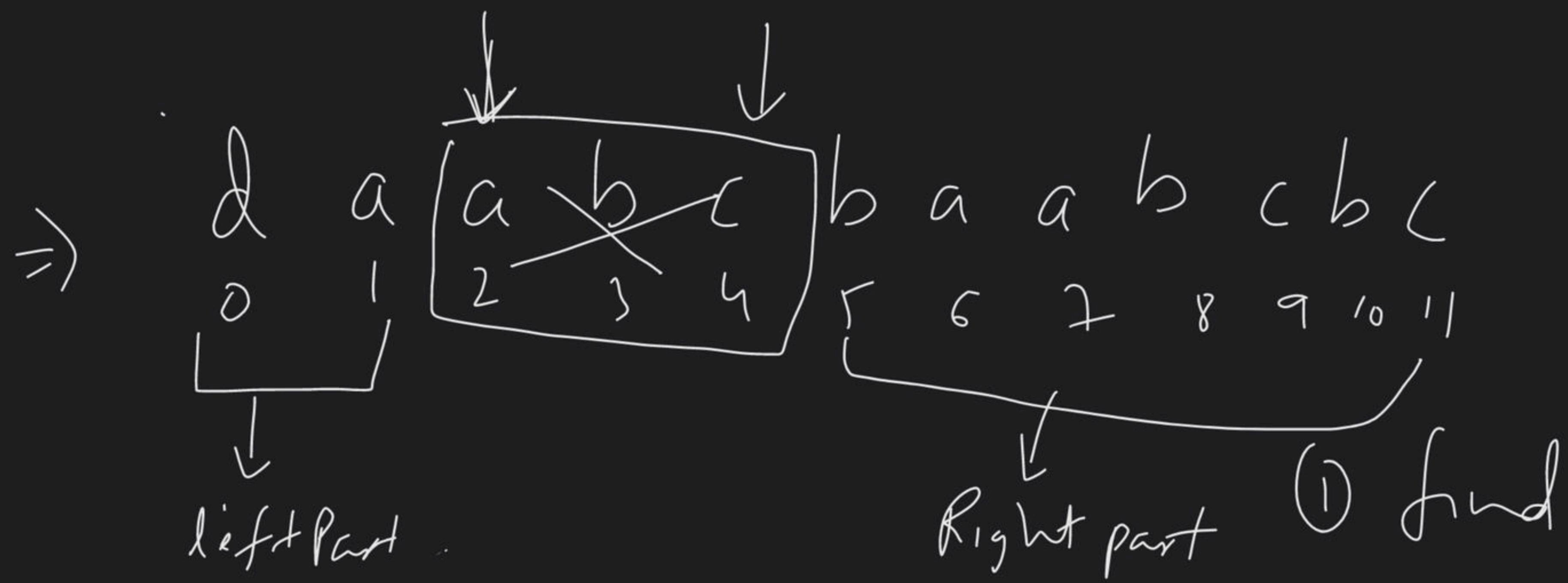
while {
    if (s.find(part) != string::npos) {
        s.erase(s.find(part), part.length());
    }
}
  
```

Diagram illustrating the execution flow of the code. The code is enclosed in a blue box. The variable 'part' is highlighted in orange. The condition `s.find(part) != string::npos` is shown with arrows pointing to both parts of the expression. The call to `s.erase` is also highlighted in orange. The curly braces of the while loop are also highlighted in orange.

$\Rightarrow L \cdot C \rightarrow | \uparrow \downarrow \cup$

Part = "abc"





$\hookrightarrow \hookrightarrow$ Left part (concat) Right part \cup

$$\Rightarrow d a \underset{=}{\cancel{b a a b c b c}} \star$$

$LP \Rightarrow S.\text{substr}(0, \text{found})$

found = 2
= 6
find(abc) → 2

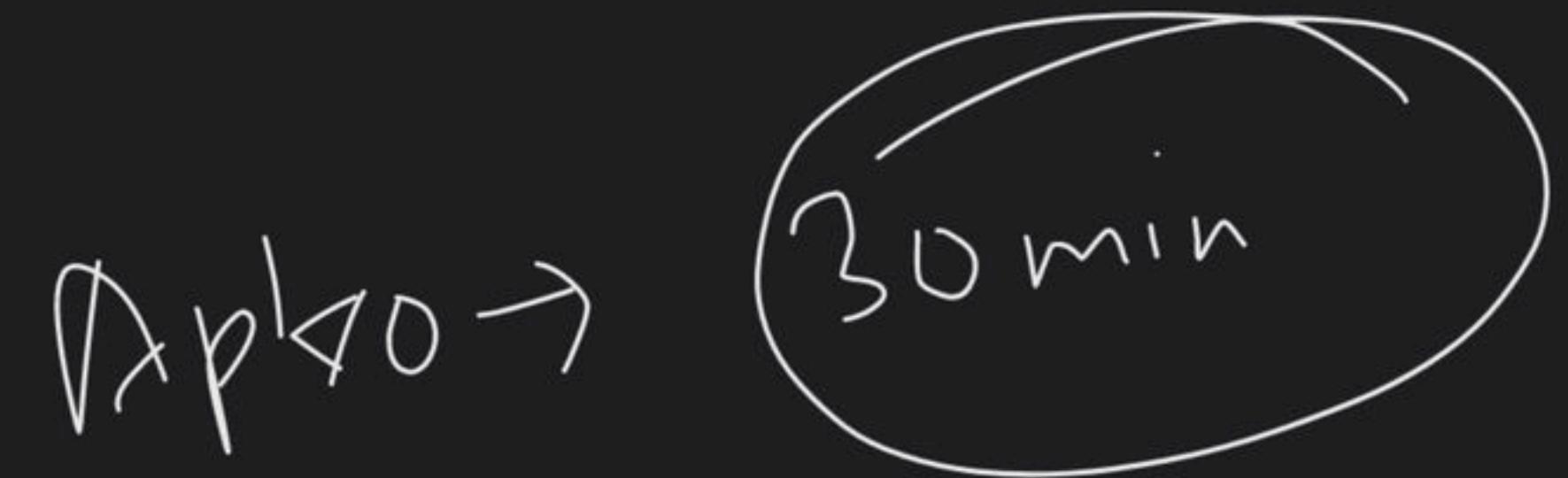
$R \Rightarrow S.\text{subst}(\text{found} + \text{Part}, \text{size}(), S.\text{size}())$



$\Rightarrow \sqrt{\tau} \rightarrow$ Sol \Rightarrow

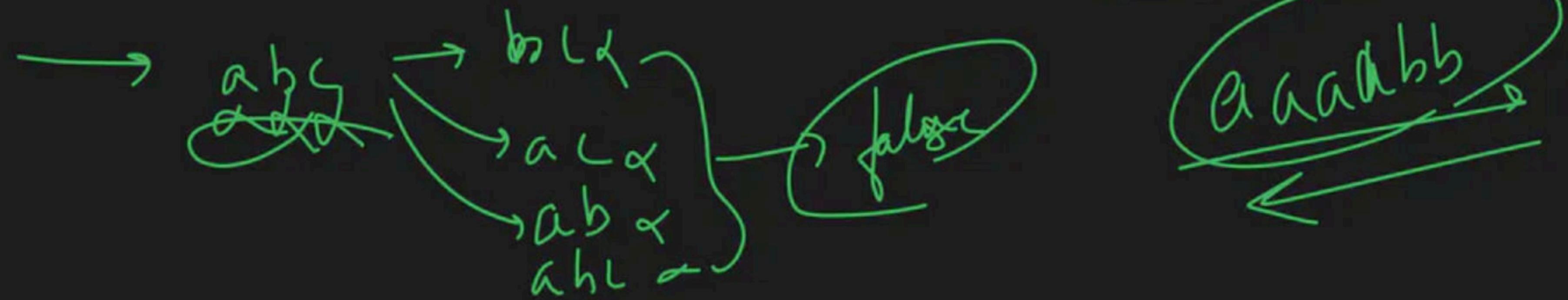
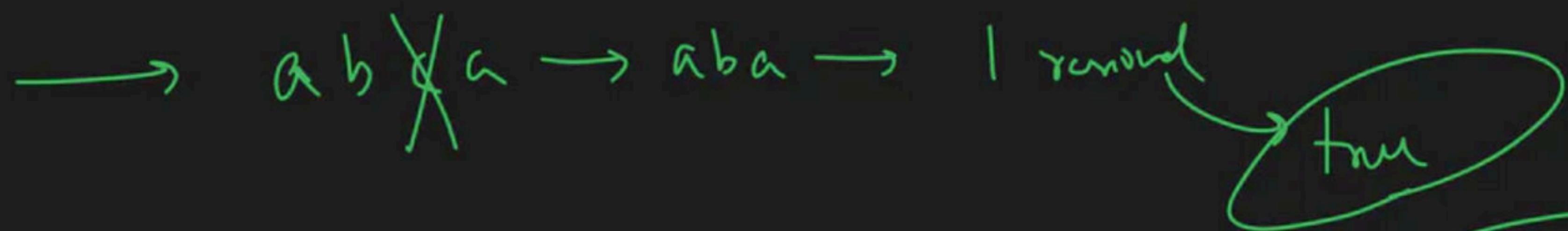
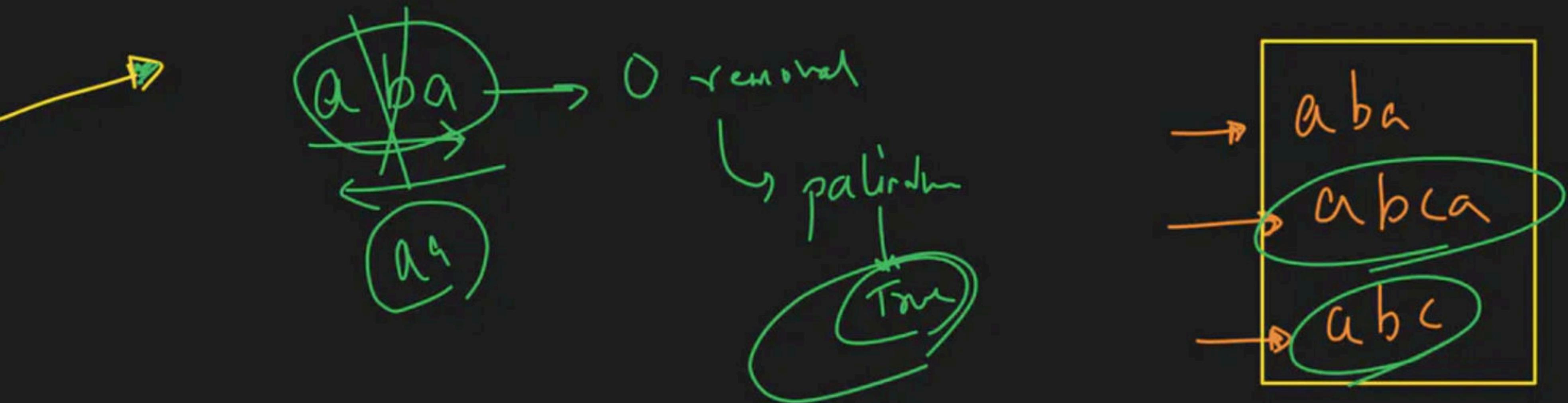


5 min
=



Apkao \rightarrow





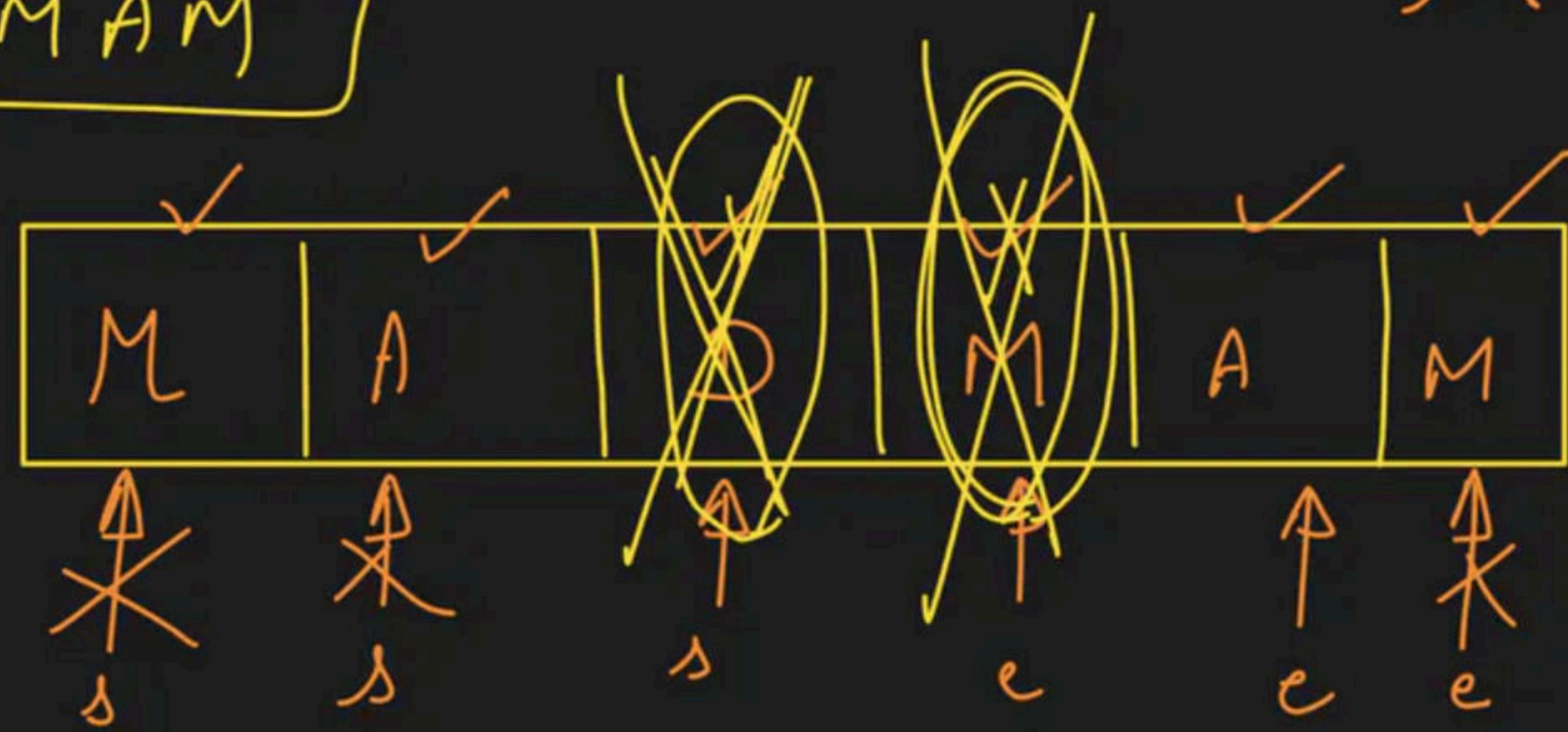
Palindrome

str

new
st

2 pointers

→ MADAM

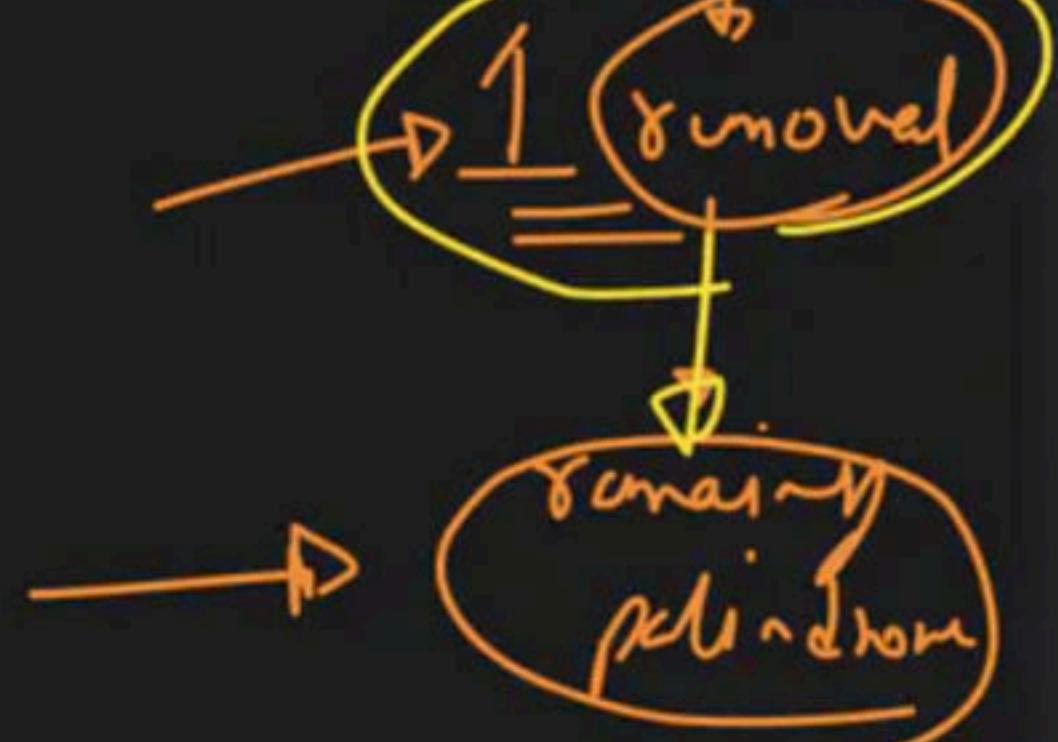


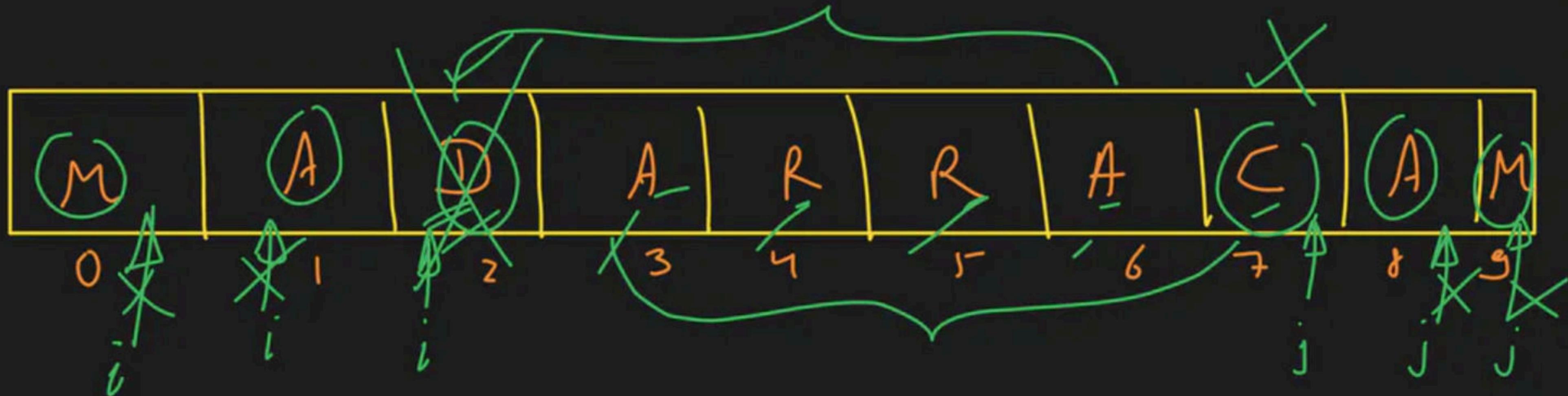
~~out~~

Icaron

equal
↓
b++
c--

not equal





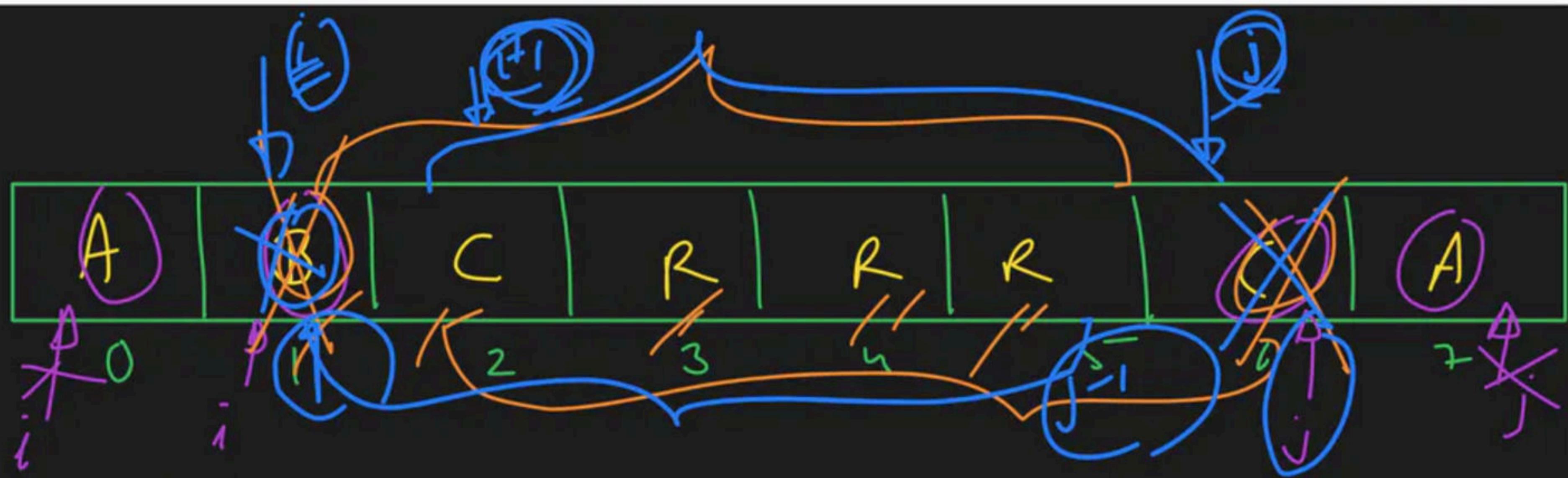
$$s[0] = -s[j] \rightarrow \text{true} \rightarrow i++ , j--$$

$$s[1] = -s[8] \rightarrow \text{true} \rightarrow i++ , j--$$

Overall as False

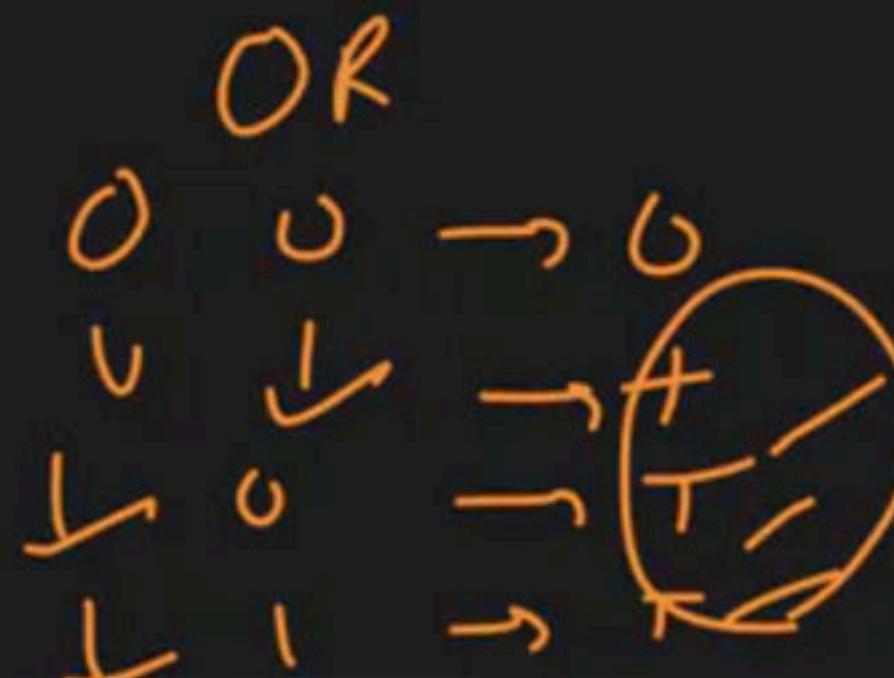
$$s[2] = -s[7] \rightarrow \text{false} \rightarrow 1 \text{ removal} \rightarrow \text{Kiko removed from } \xrightarrow{\text{D runout}} \text{ARRAC}$$

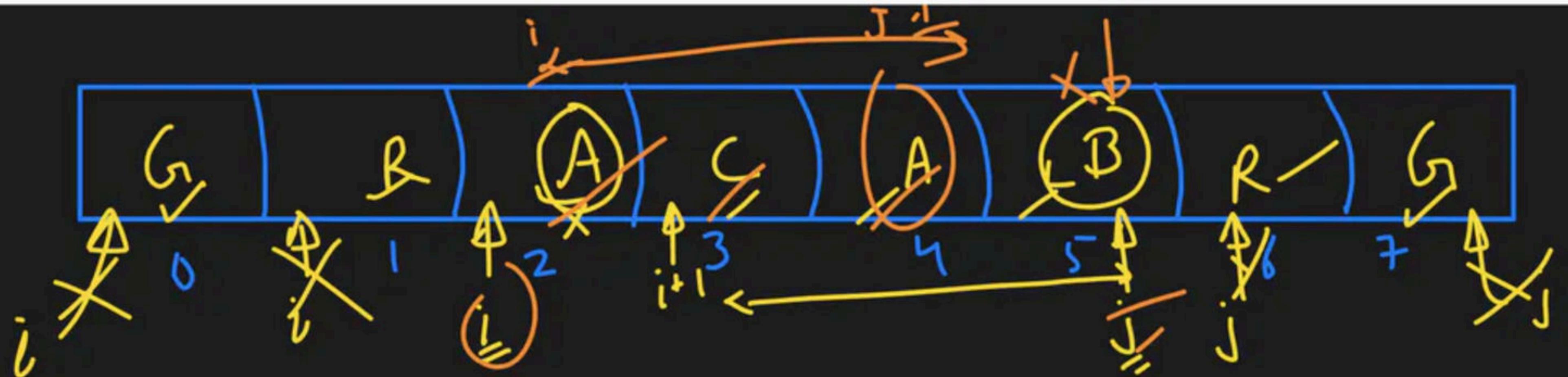
DARRA \rightarrow D



$s[0] = s[7] \rightarrow \text{true} \rightarrow i++ j--$

$s[1] \Rightarrow s[1] \rightarrow \text{false} \rightarrow$ *lremove* \rightarrow *B remove* \rightarrow *true*
 $\text{OR} \rightarrow$ *lremove* \rightarrow *OR* \rightarrow *CRFR* \rightarrow *find Ans*





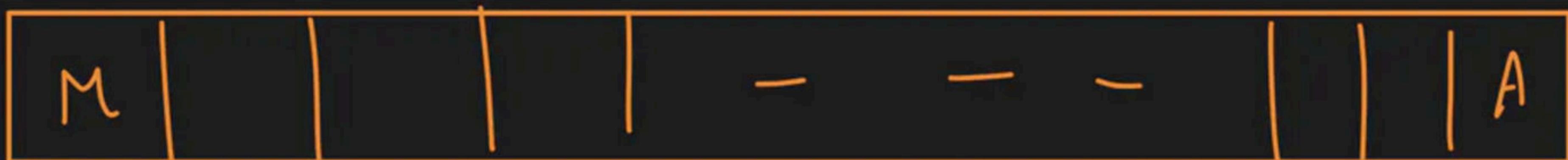
$s[0] = s[7] \rightarrow$ true $\rightarrow i++ \quad j--$

$s[1] = s[6] \rightarrow$ true $\rightarrow i++ \quad j--$

$s[2] = s[5] \rightarrow$ false
 → remove \rightarrow i^{th} character (false)
 \rightarrow remaining j^{th} character (true)

\rightarrow $i \rightarrow j \rightarrow$ array str

~~$i \rightarrow j \rightarrow$~~



$\text{check}(1 \rightarrow n-2) \rightarrow \cancel{\text{check}}(1 \dots n-2) \rightarrow O(n-1)$

$\text{check}(2 \rightarrow n-1) \rightarrow \cancel{\text{check}}(2 \dots n-2) \rightarrow O(n-2)$

$\text{while } (i < j)$
 $i <= j$

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

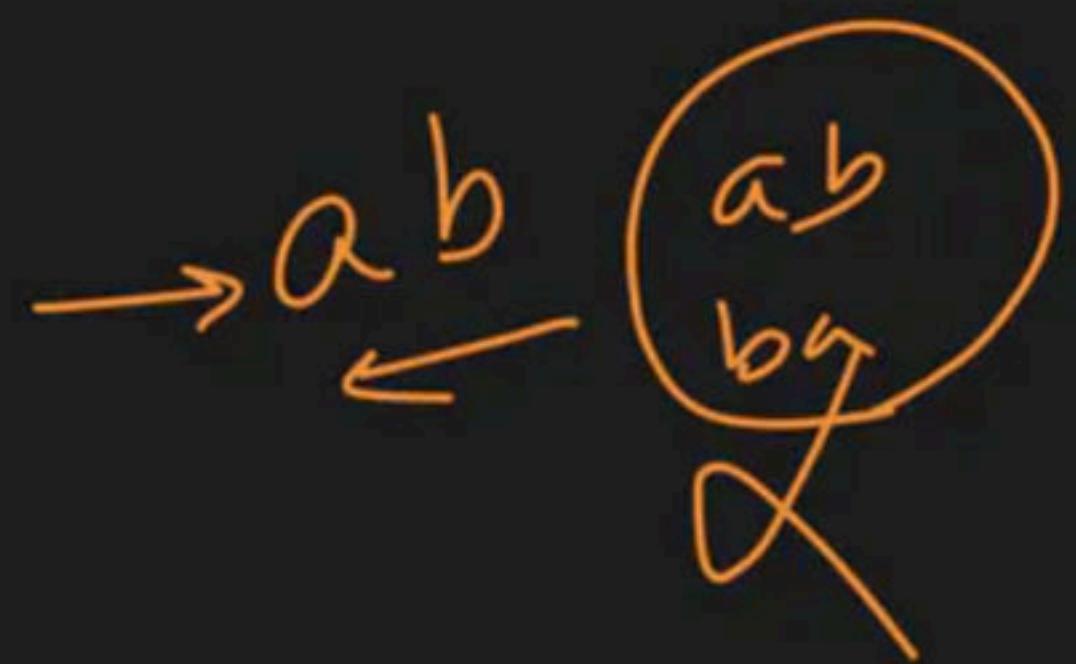
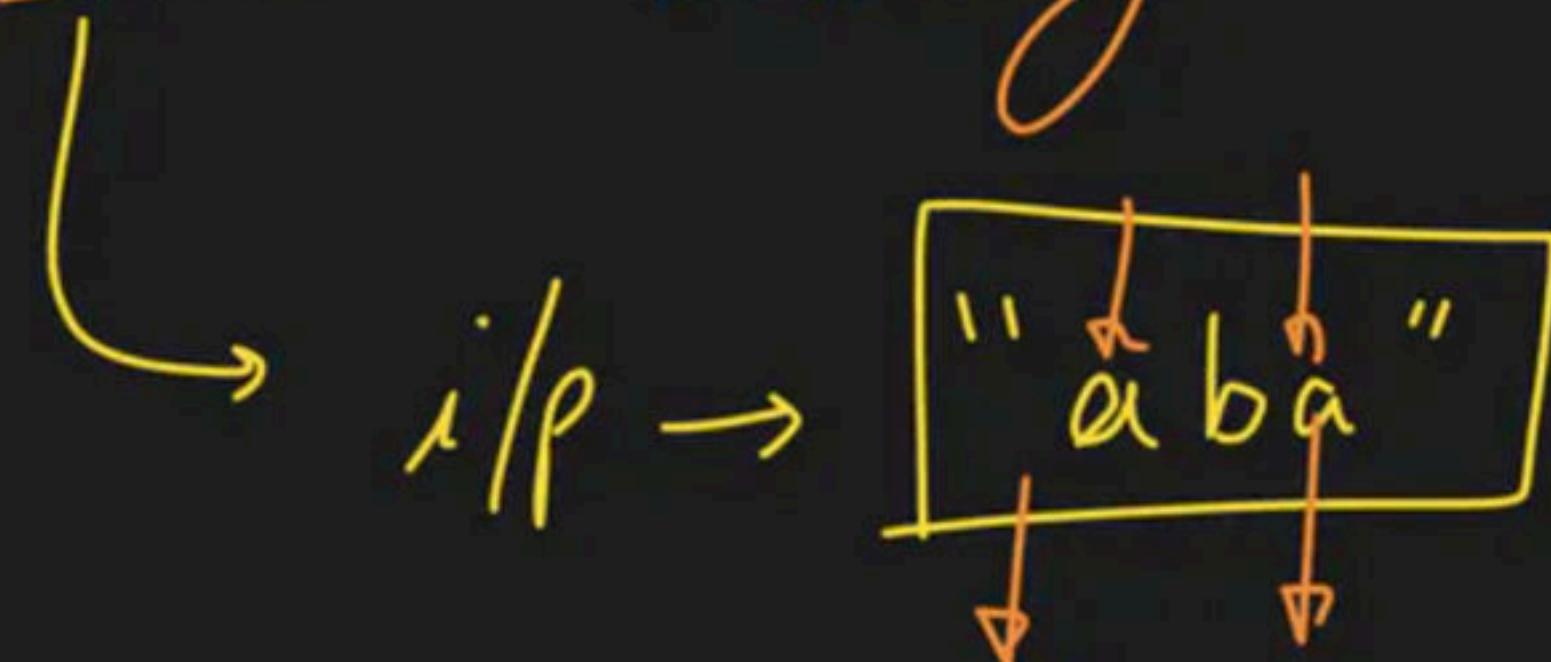
$O(n)$

Palindromic Substring :-



Count

Substr → palindrome



a	→	✓	} 2	9
aba	→	✓		
ab	→	✗		
ba	→	✗		
a	→	✓		aa
b	→	✓		=

Approach:-

Brute force

find all ~~Sub string~~ $\rightarrow O(n^2)$

~~check palindrom~~ $\rightarrow O(n)$

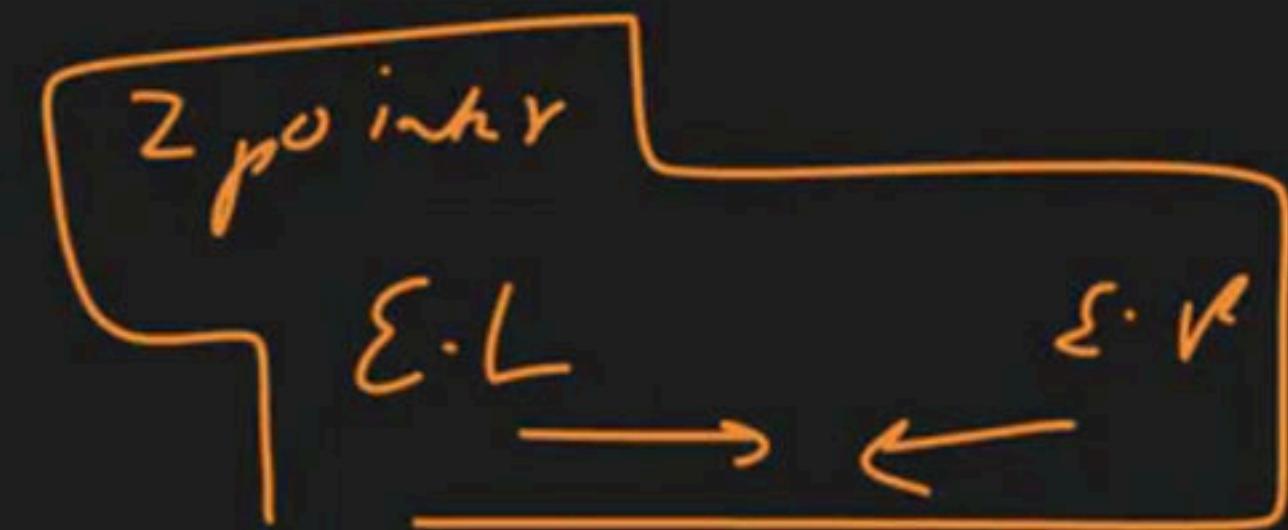
Count

a b c d

aa
cc

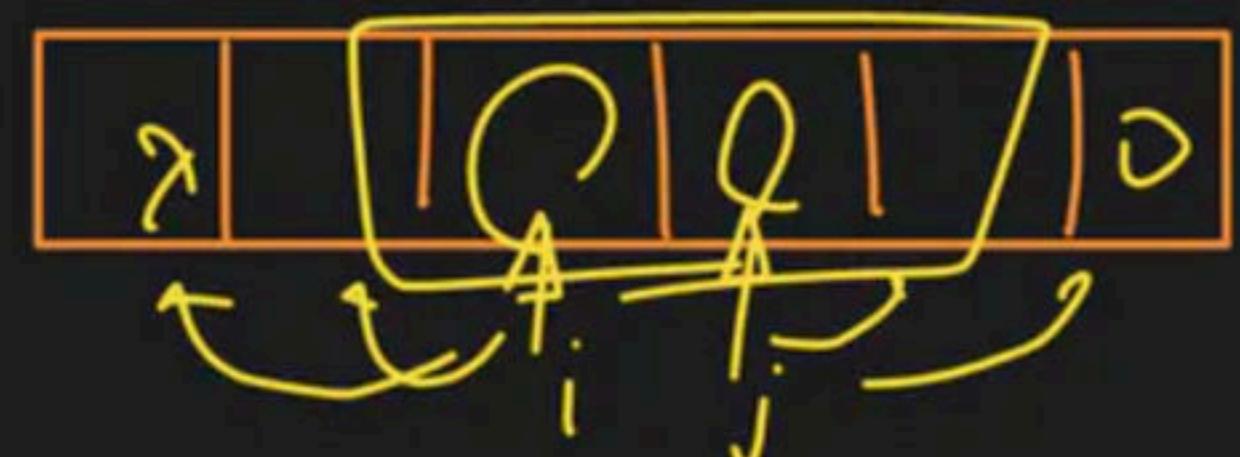
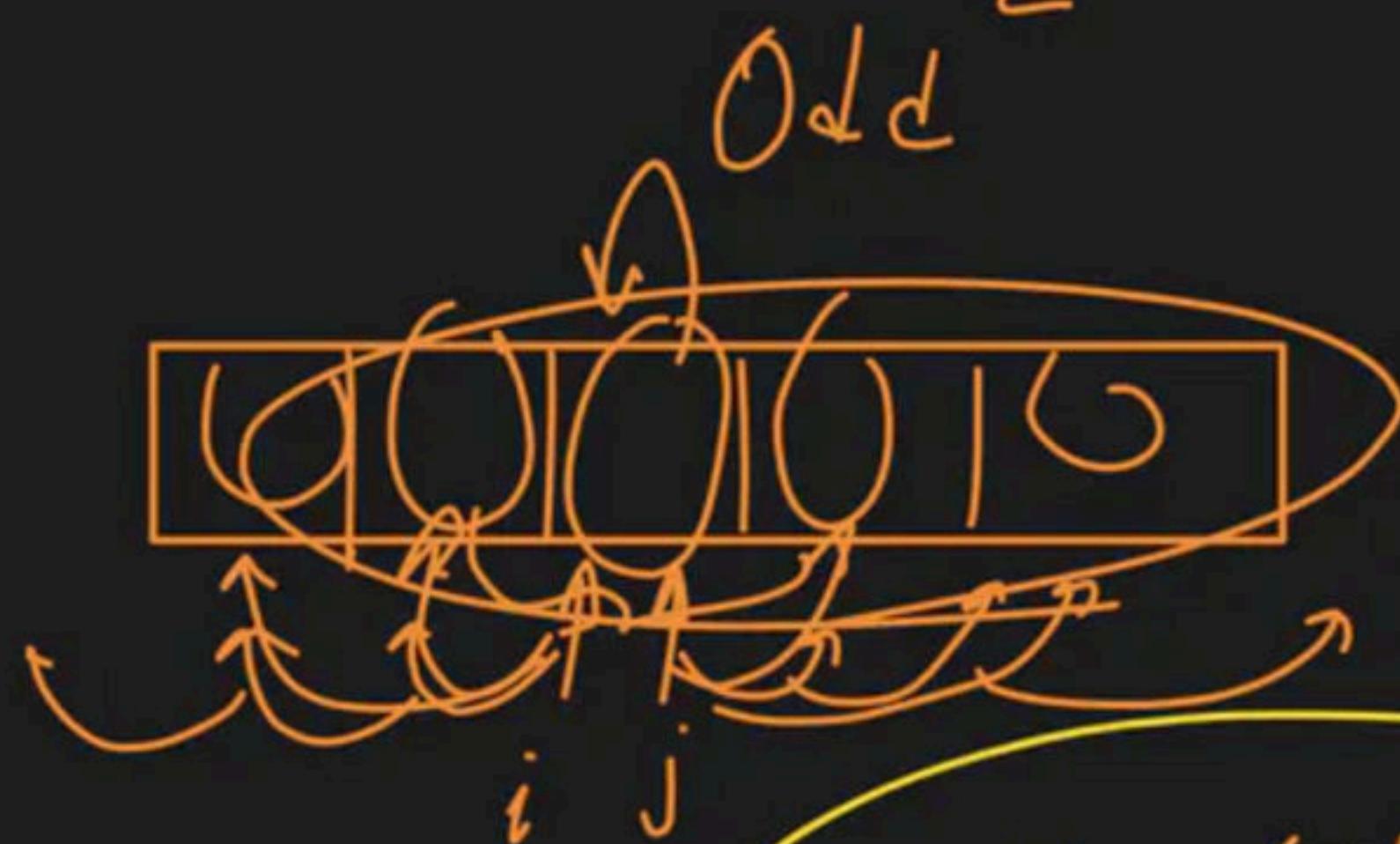
Actual Solution :-

Palindrome



Odd

Even



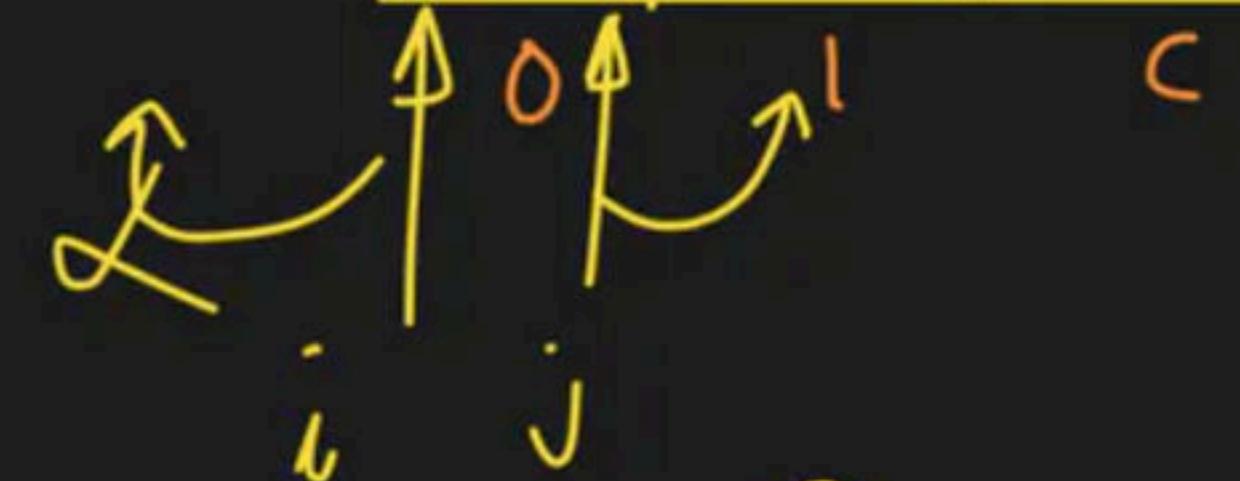
match → ~~count~~ count

not match - 1 → no need to
compare further

P

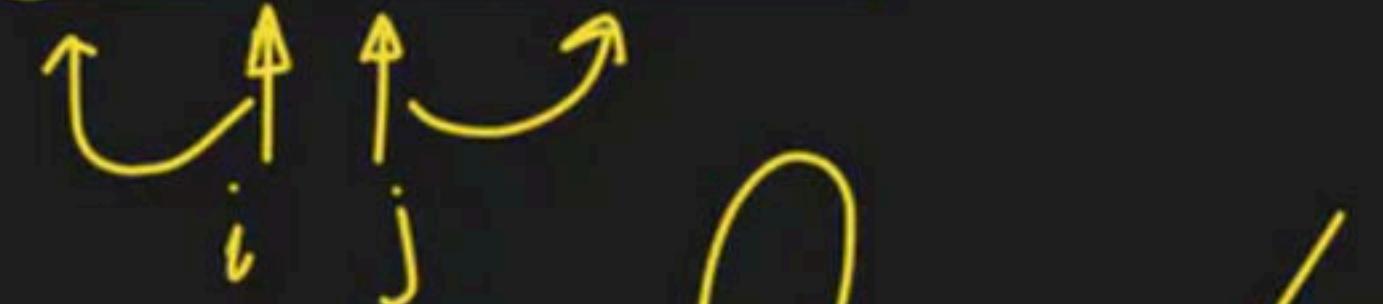
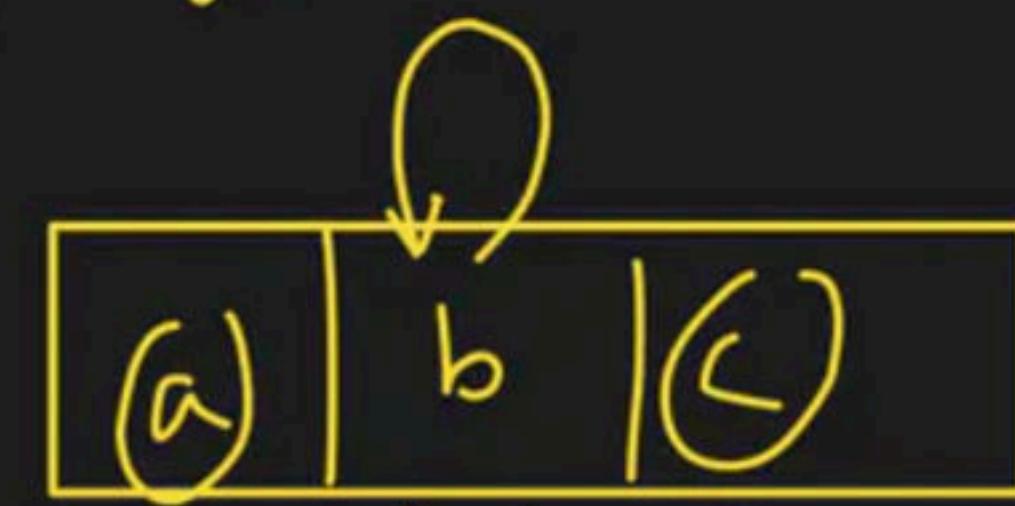
match

odd

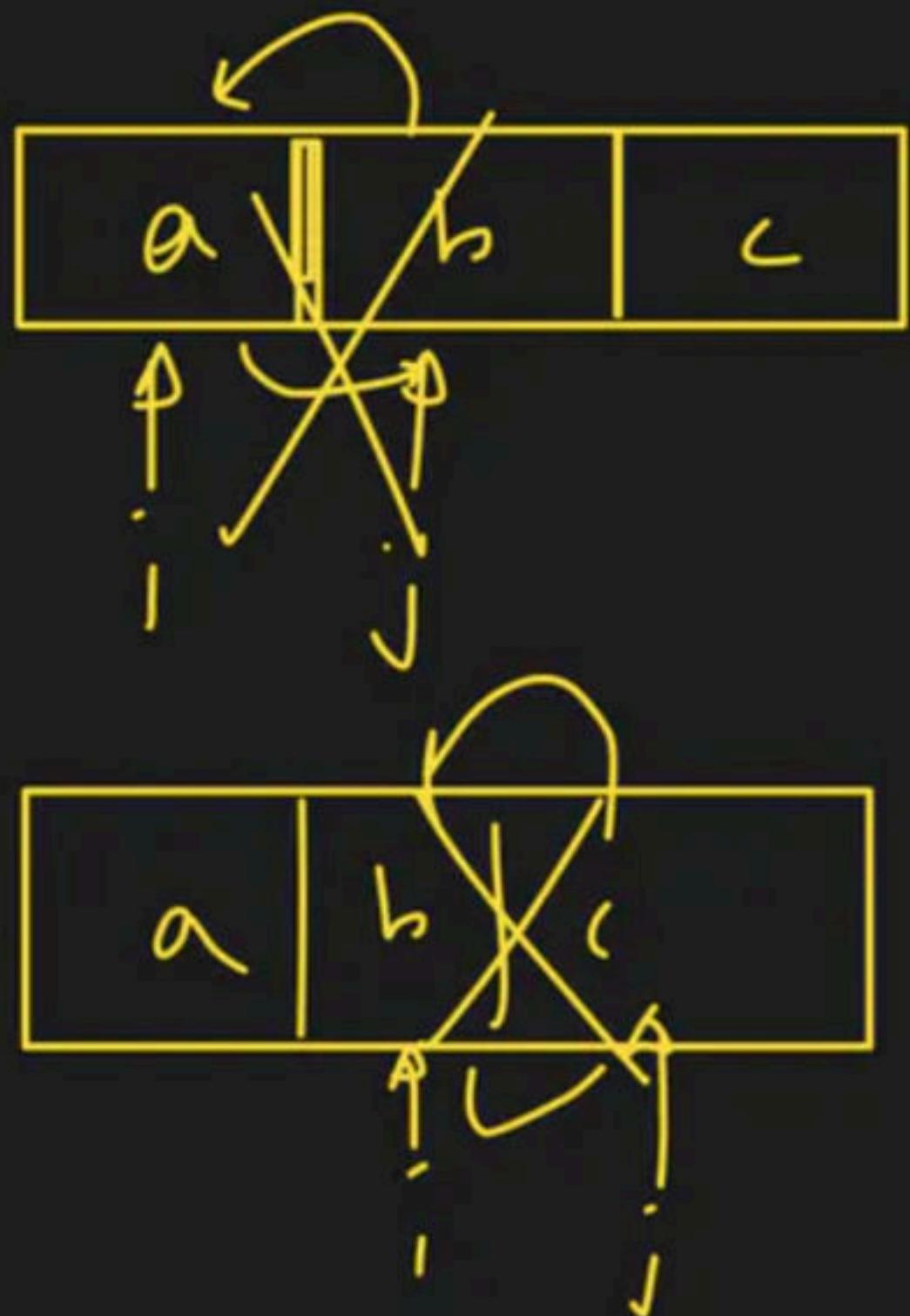


Count = 1 + 1 + 1

Count = 3



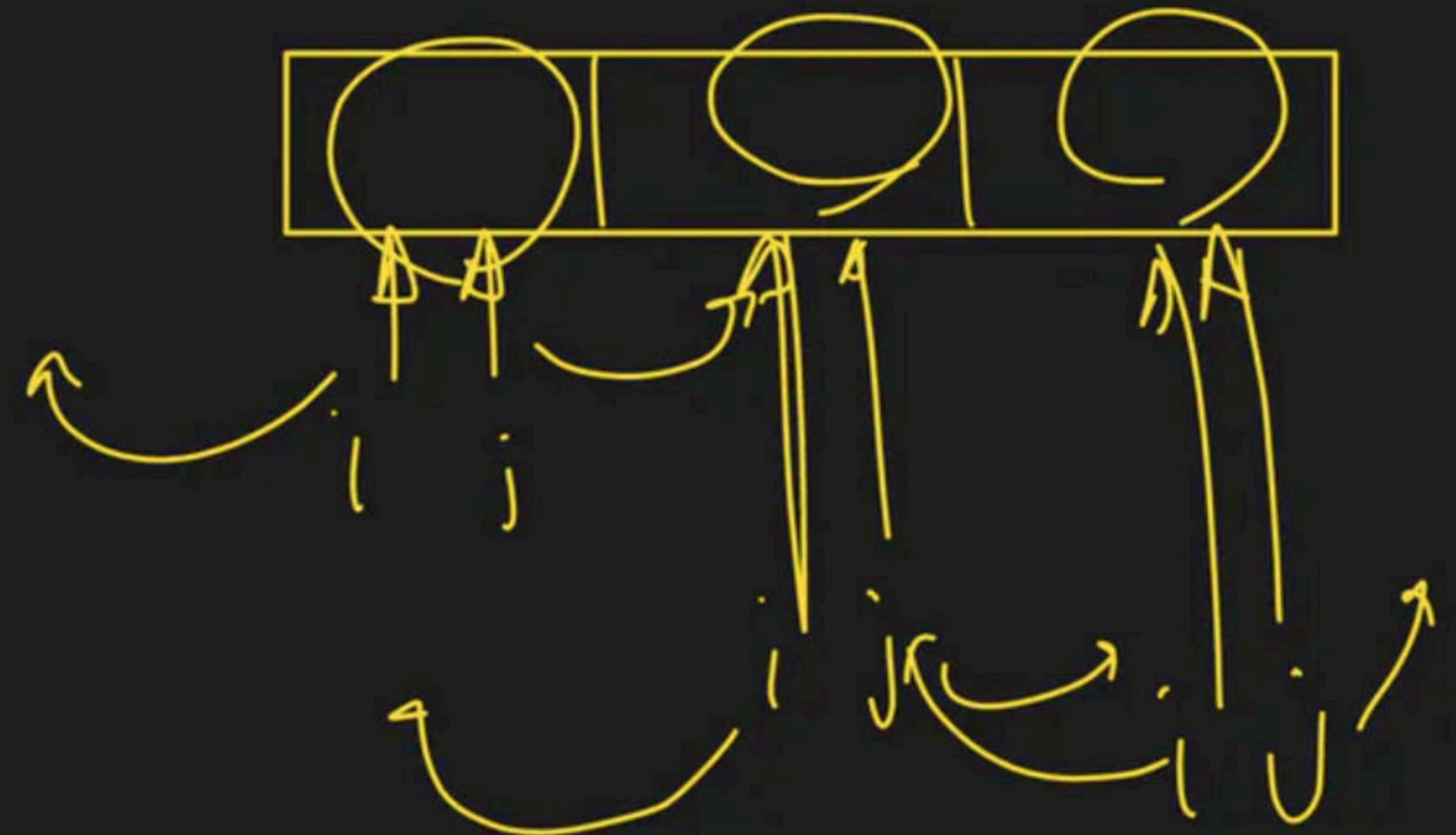
Even



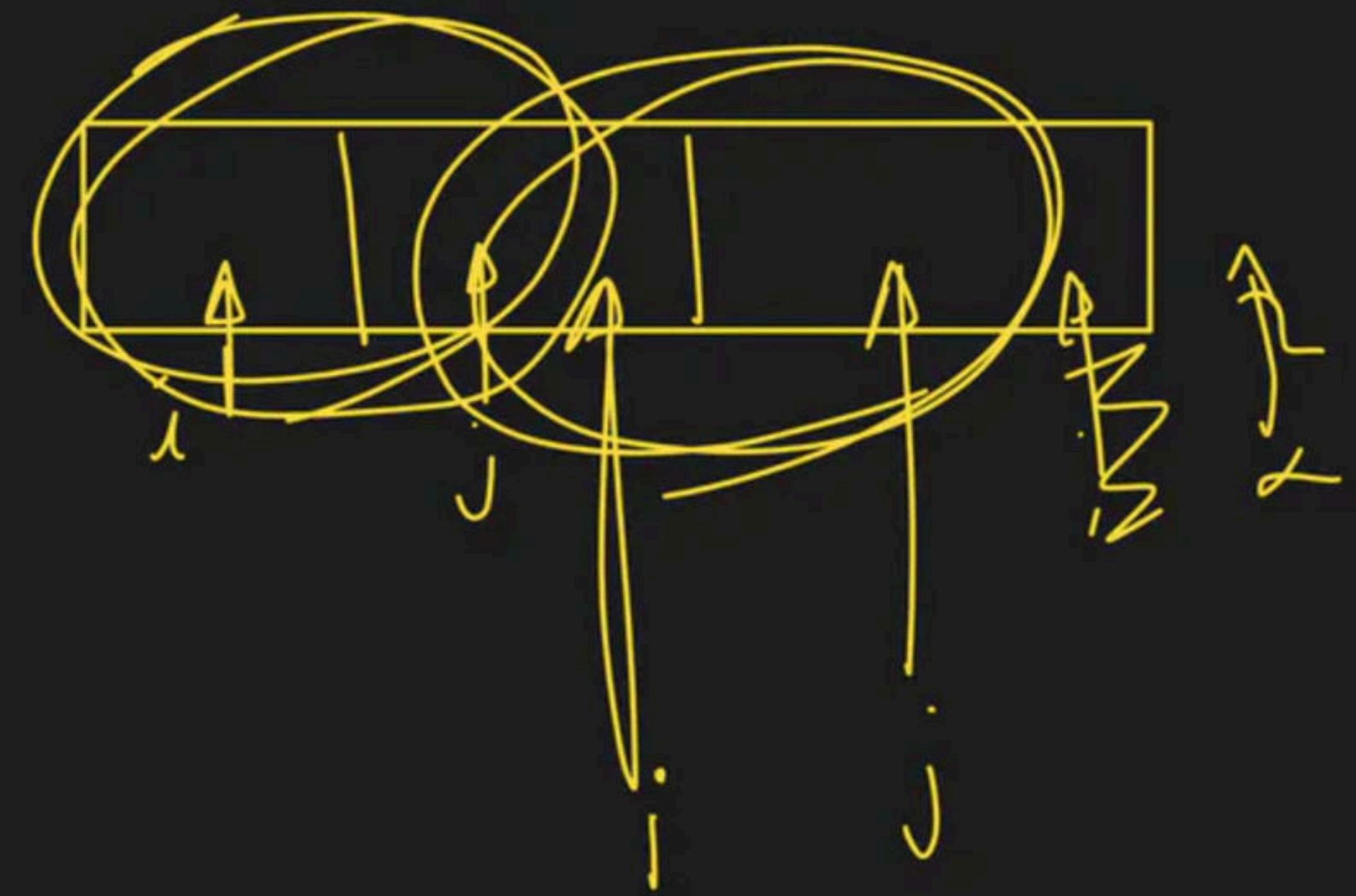
cout = 0

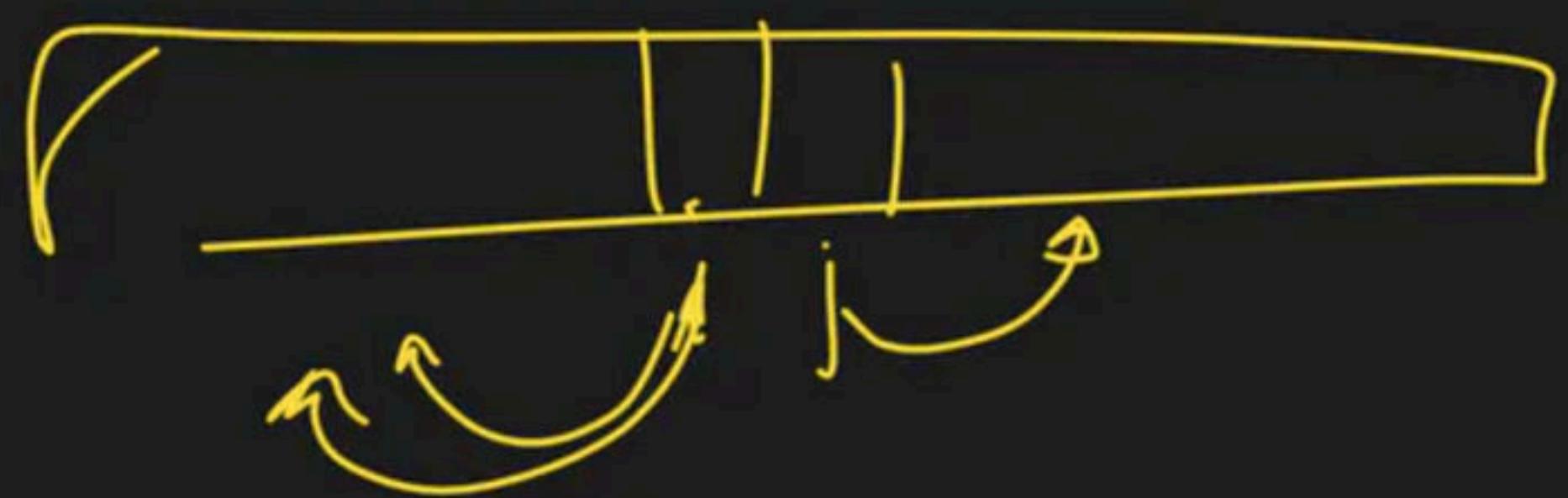
Total $\rightarrow 3 + 0$
cout
= 3

odd

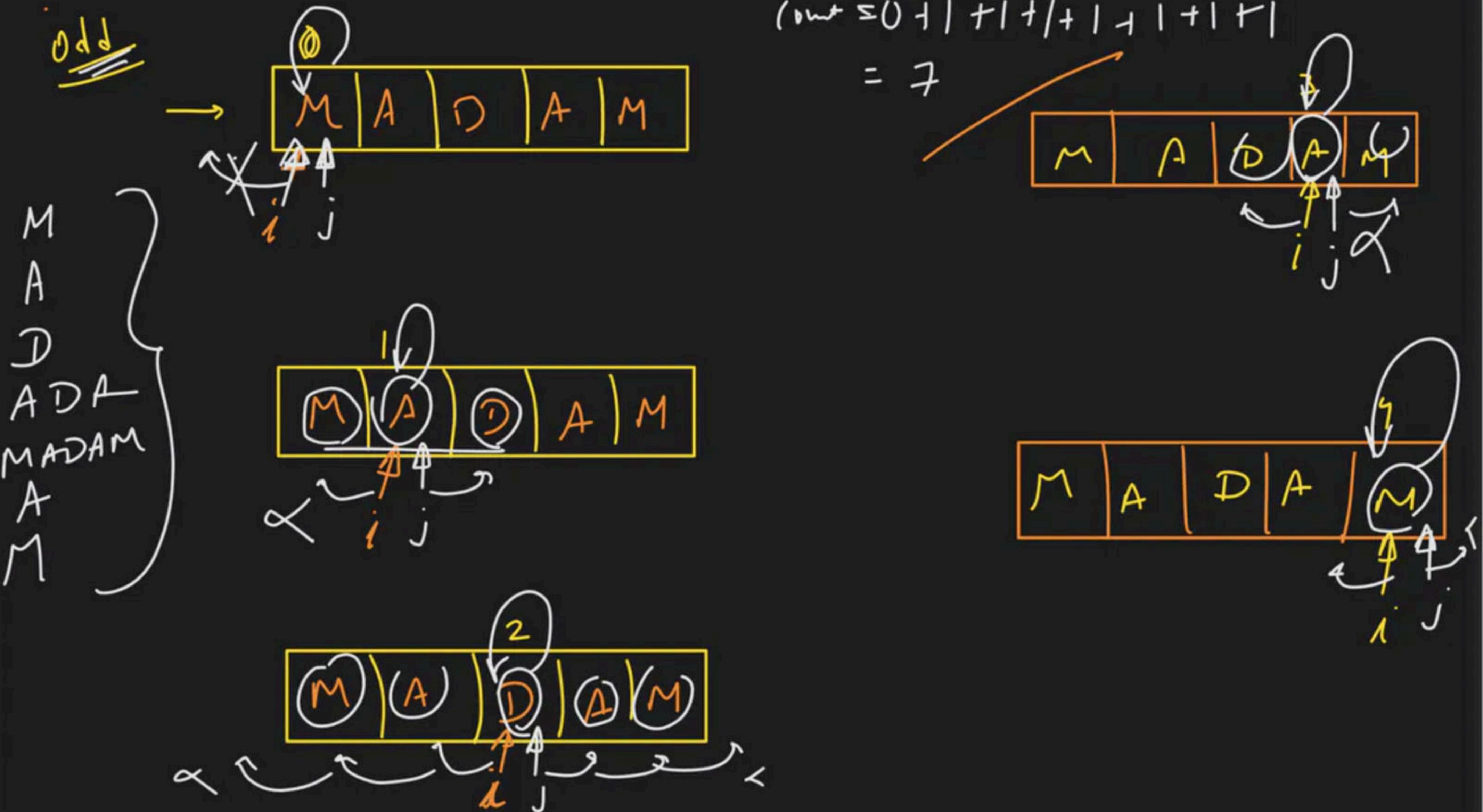


Even ~

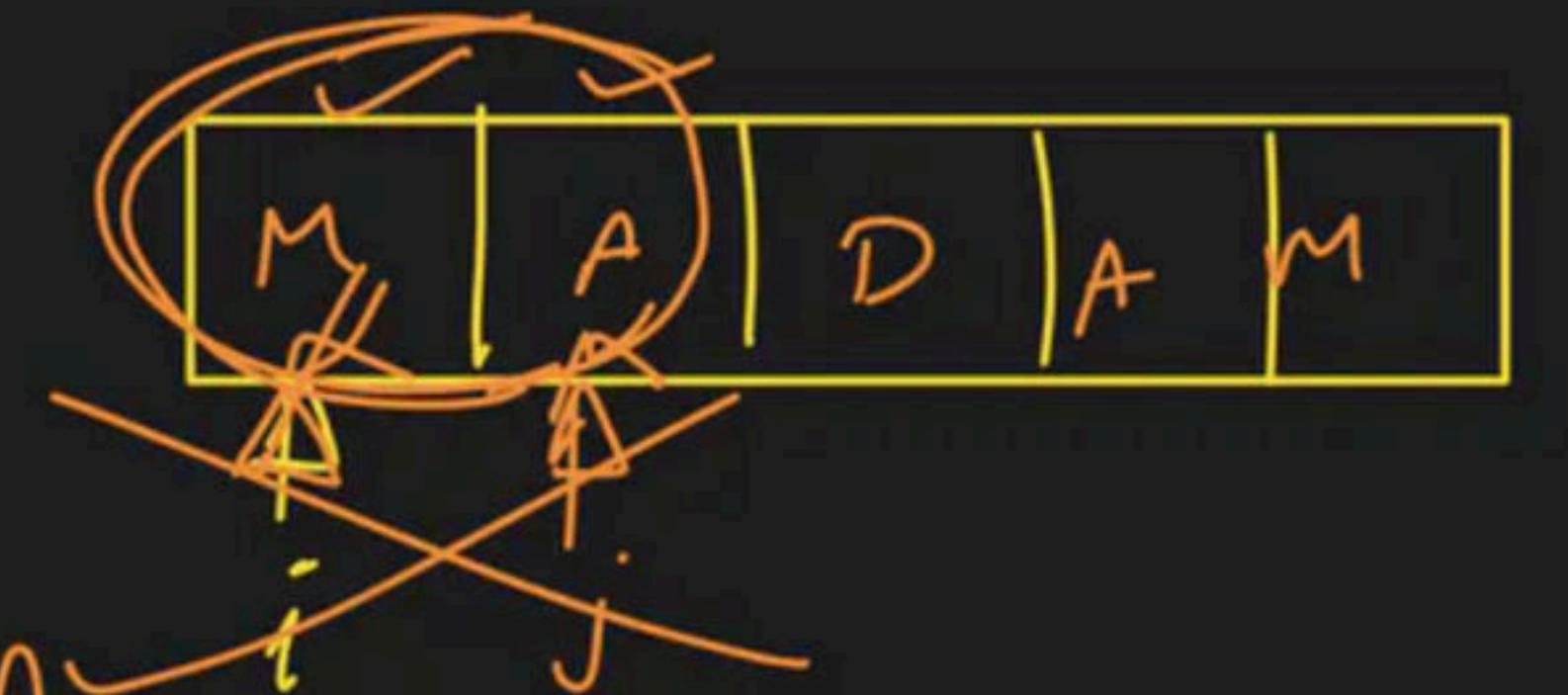








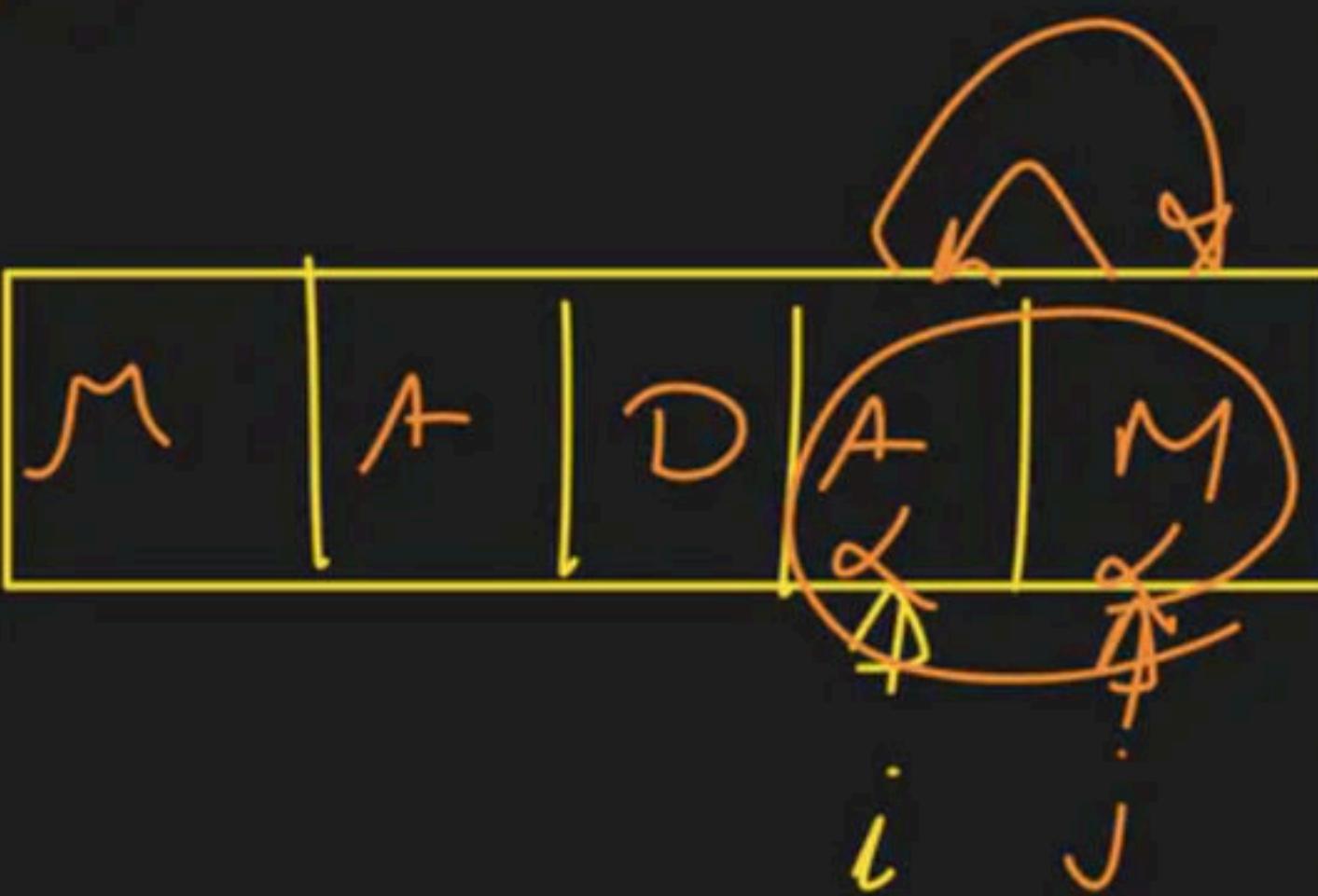
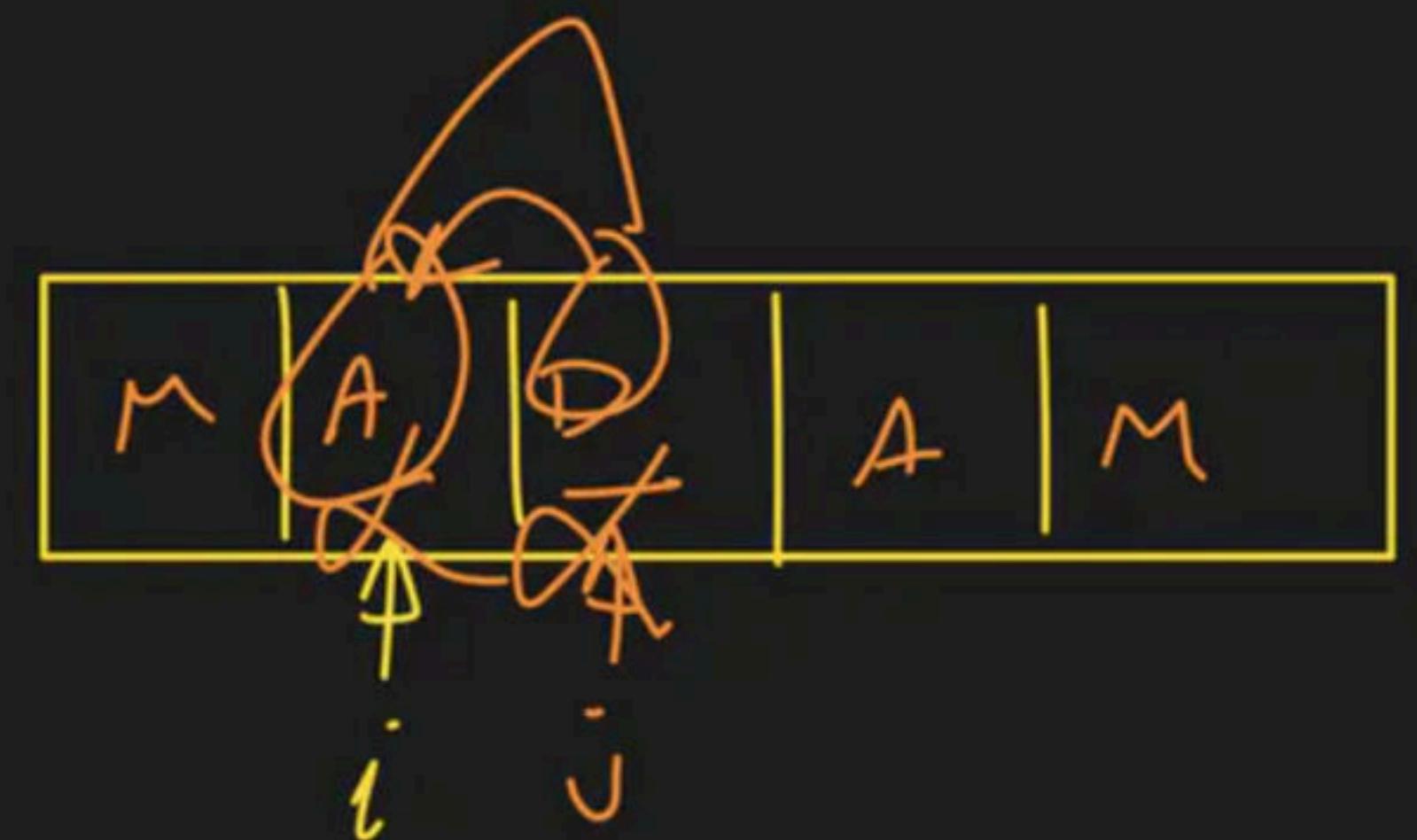
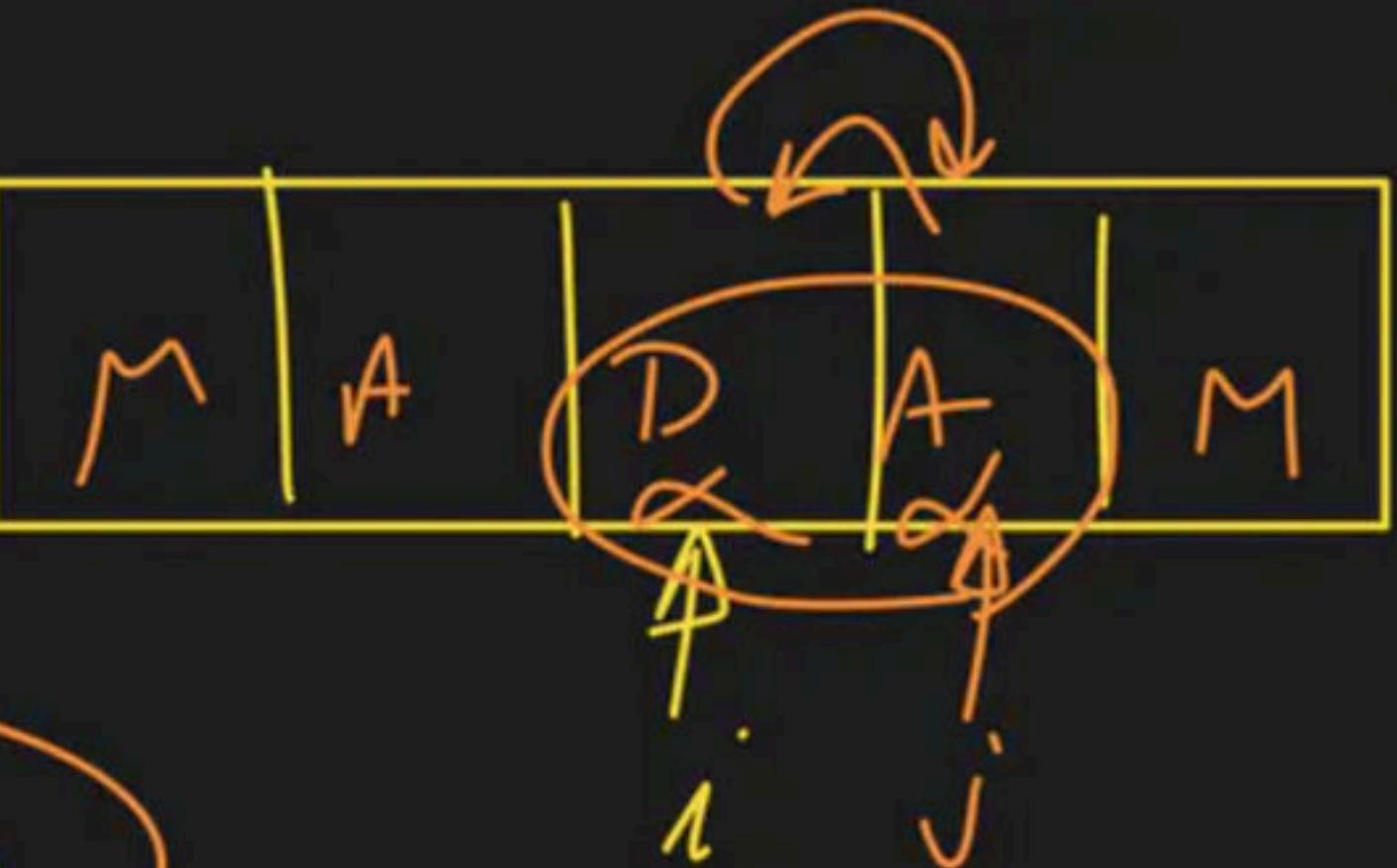
Even



$$M'_i = A$$

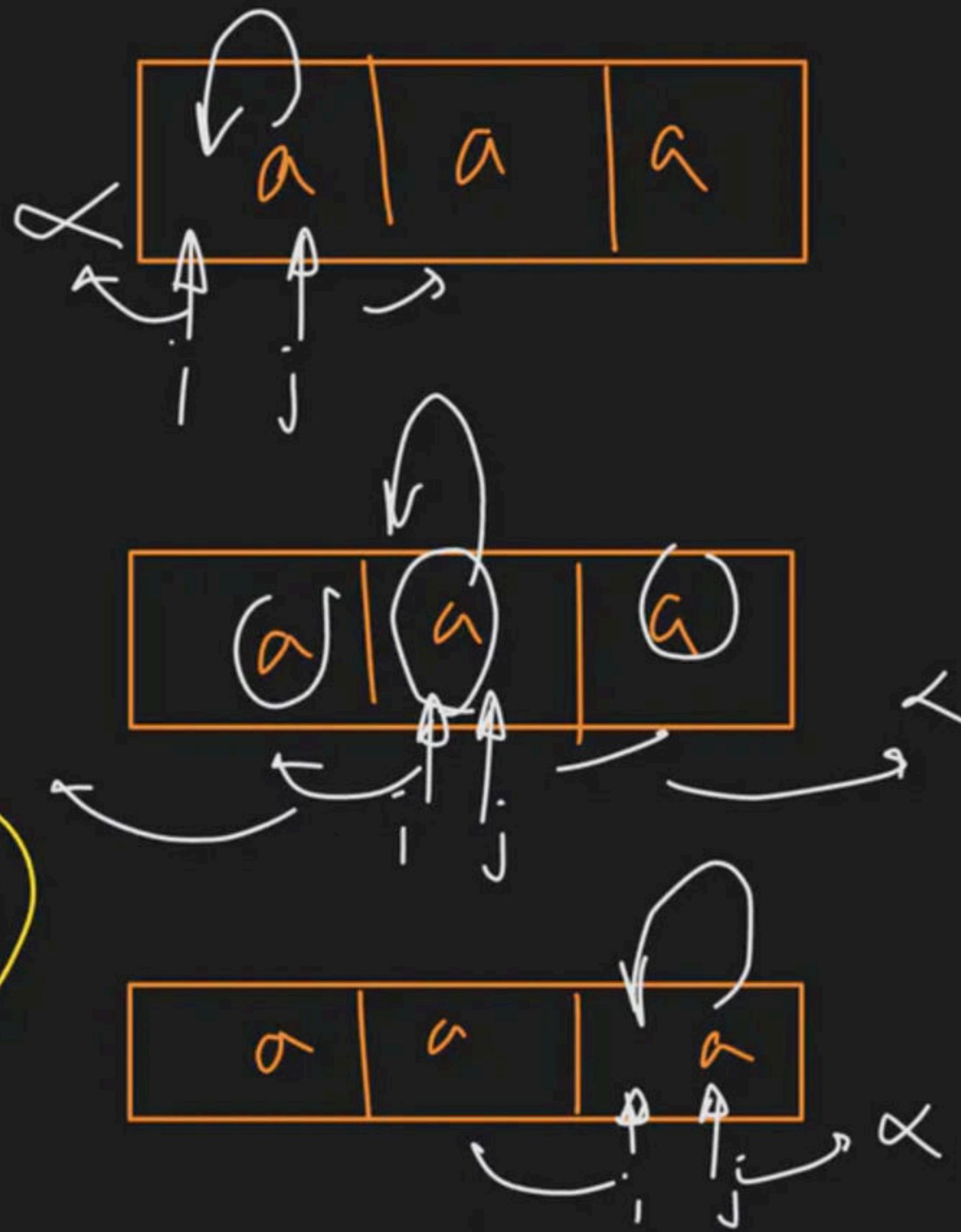
$2 \rightarrow$ Even

$(out = 0)$

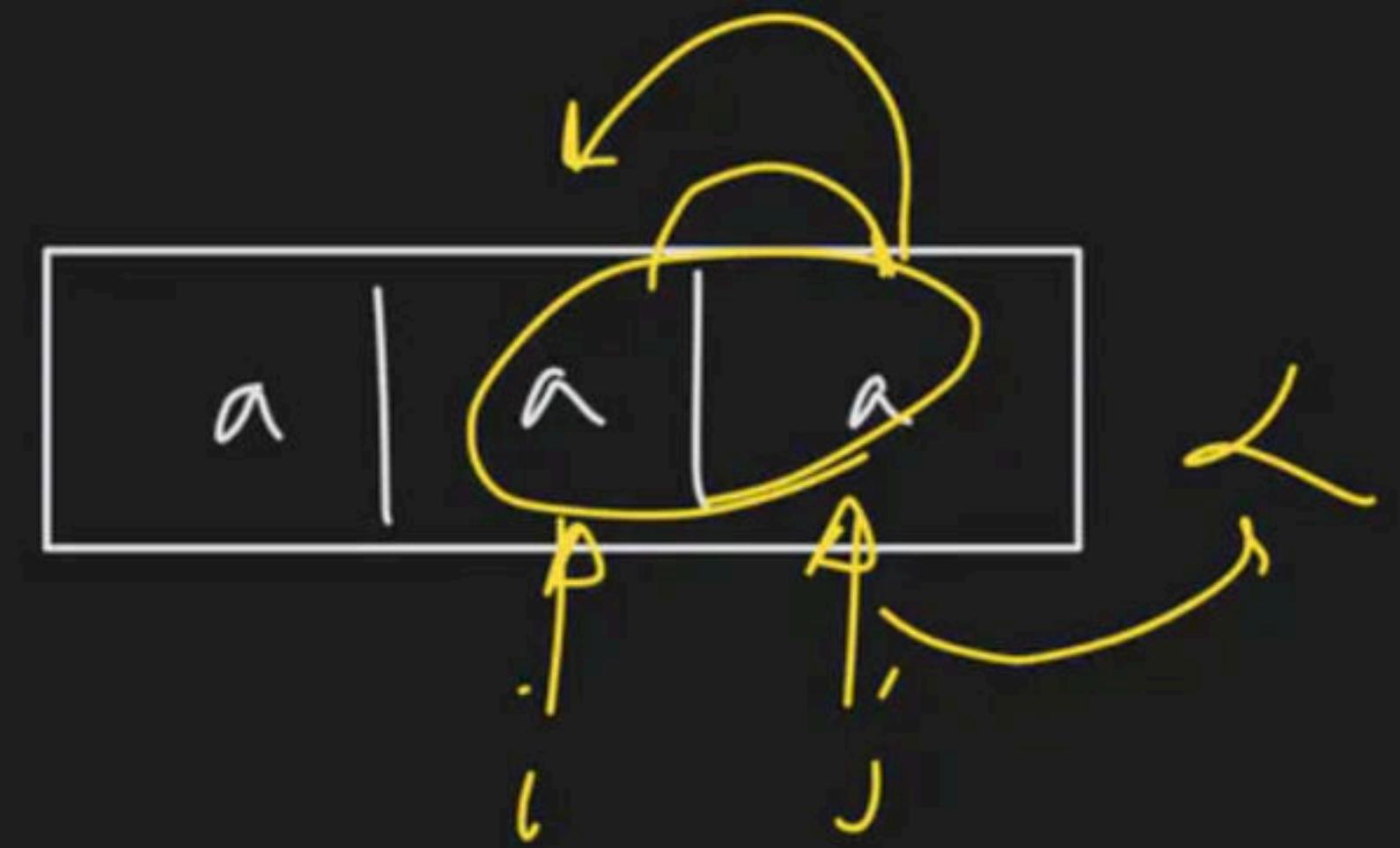
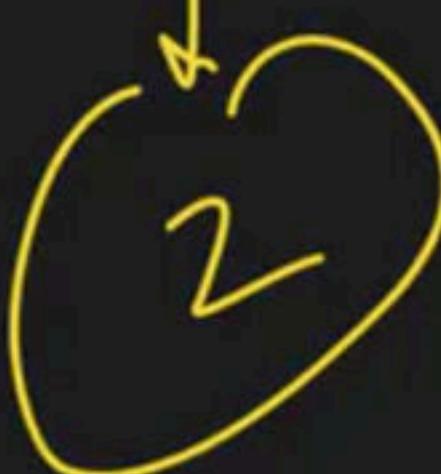
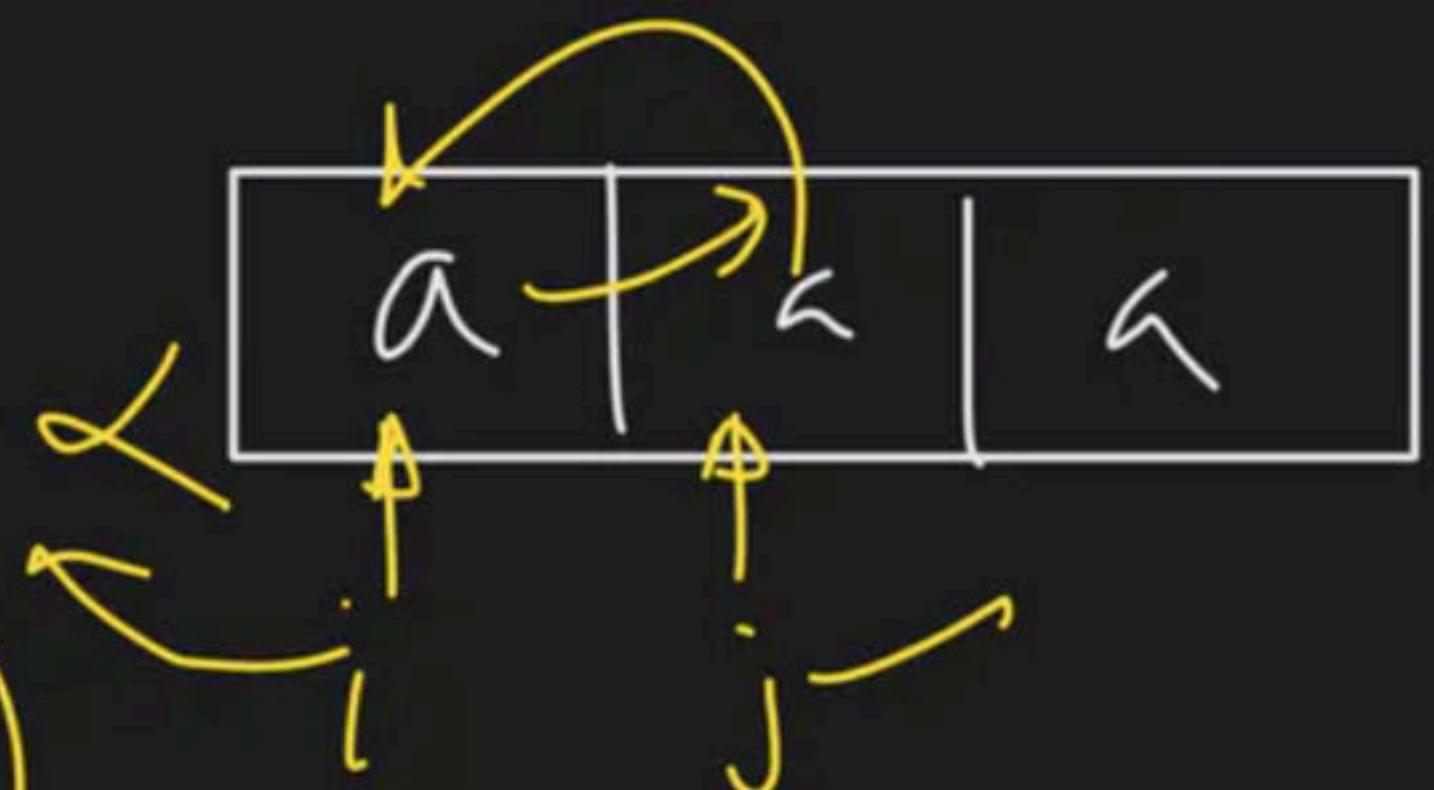


odd

a
a
aaa
a

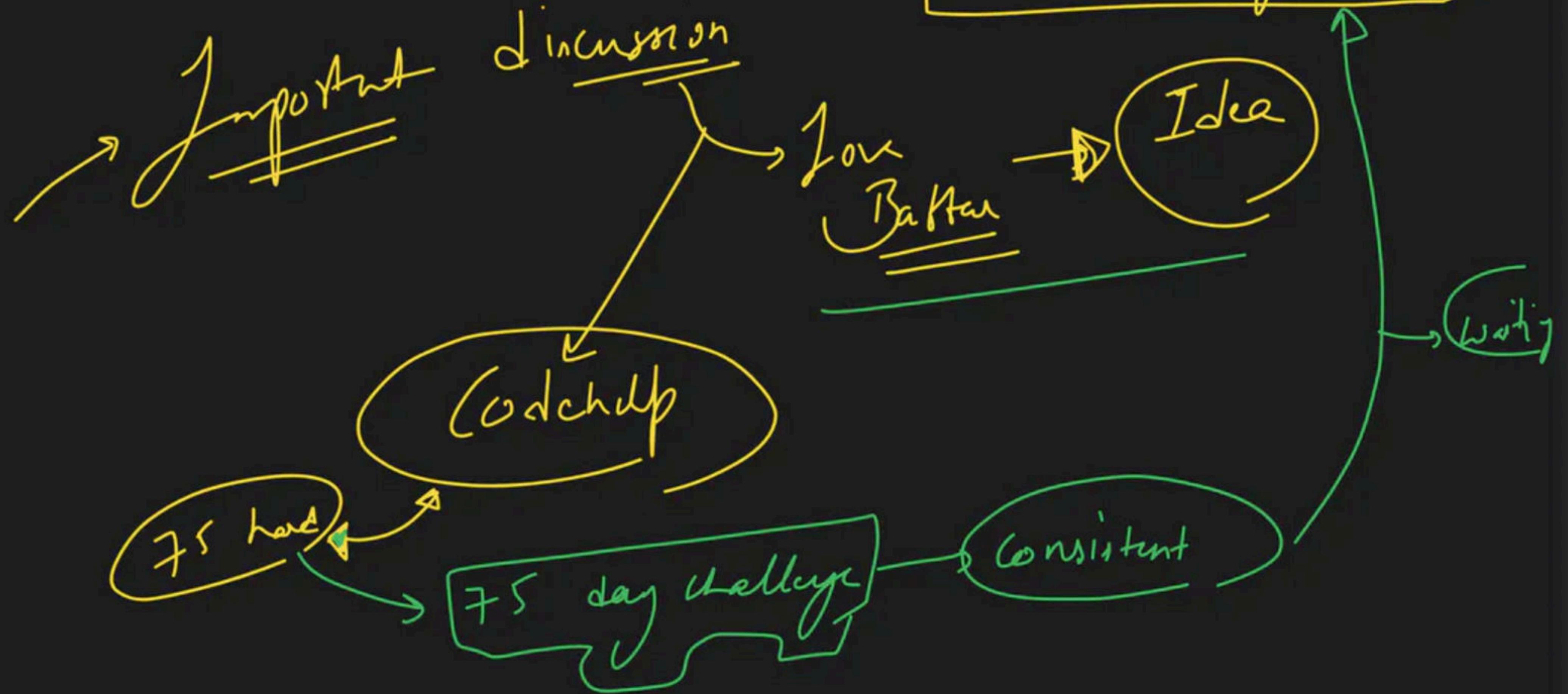


~~Even~~

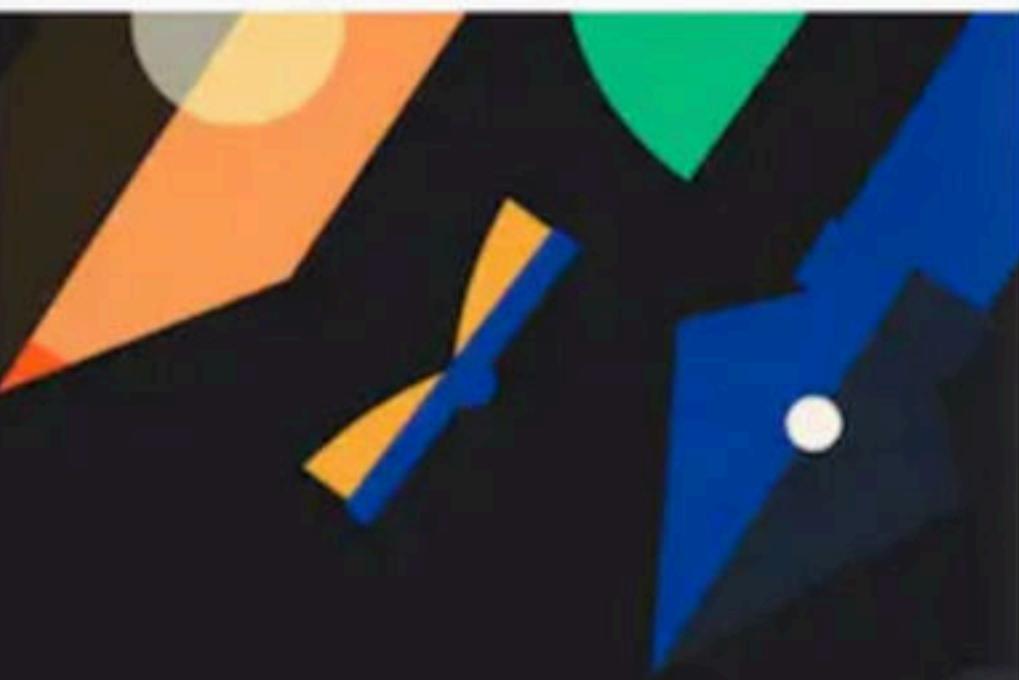


Solution \rightarrow





lou bahtan >@gmail.com



Char Arrays & Strings Class 3

Special class

→ Strings

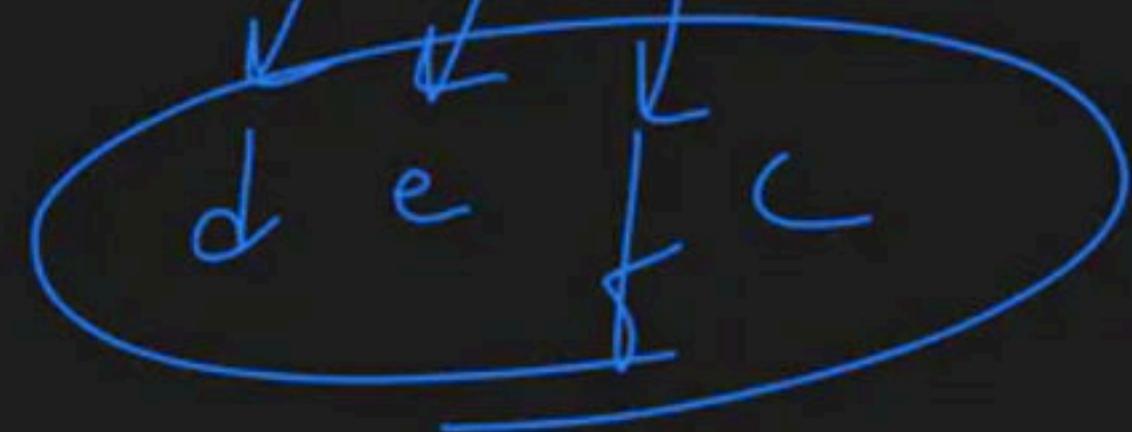
Char Array



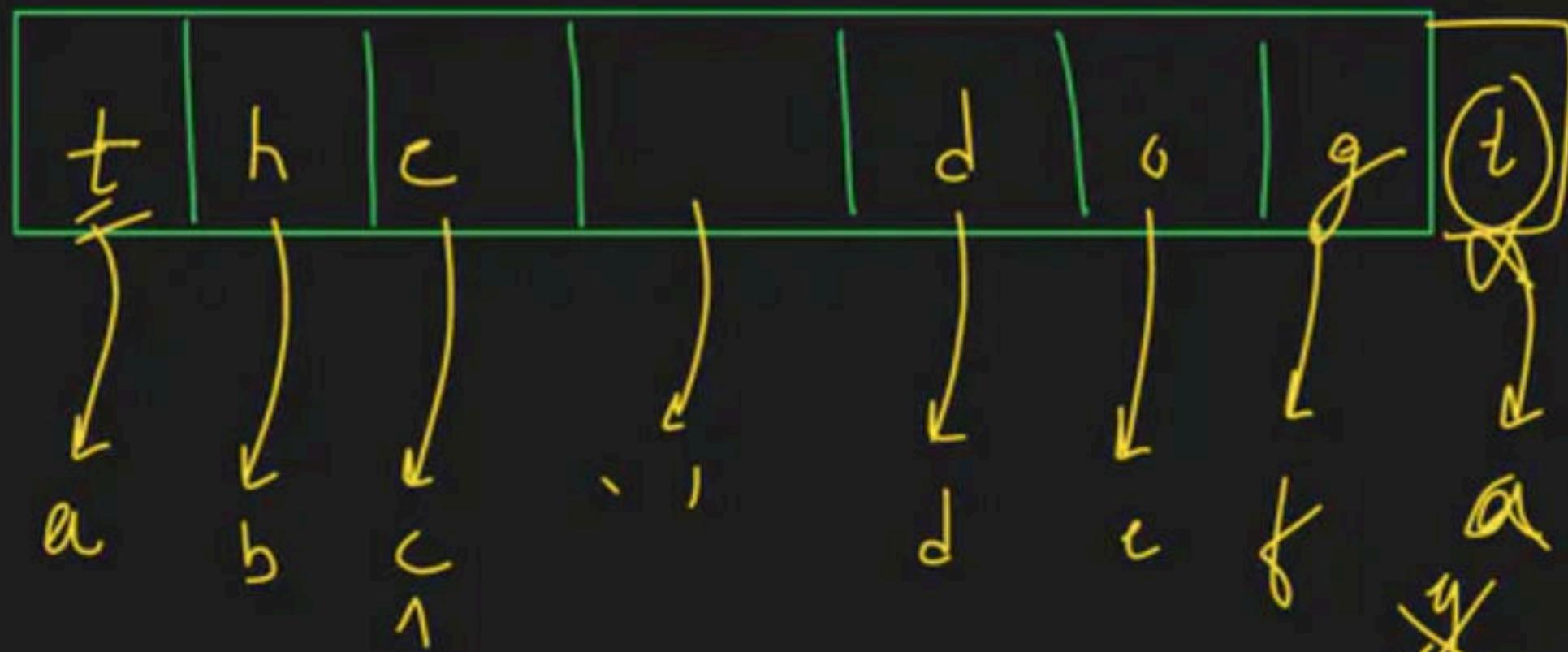
→ Questions Solve :-

map:

d	o	g	c
---	---	---	---

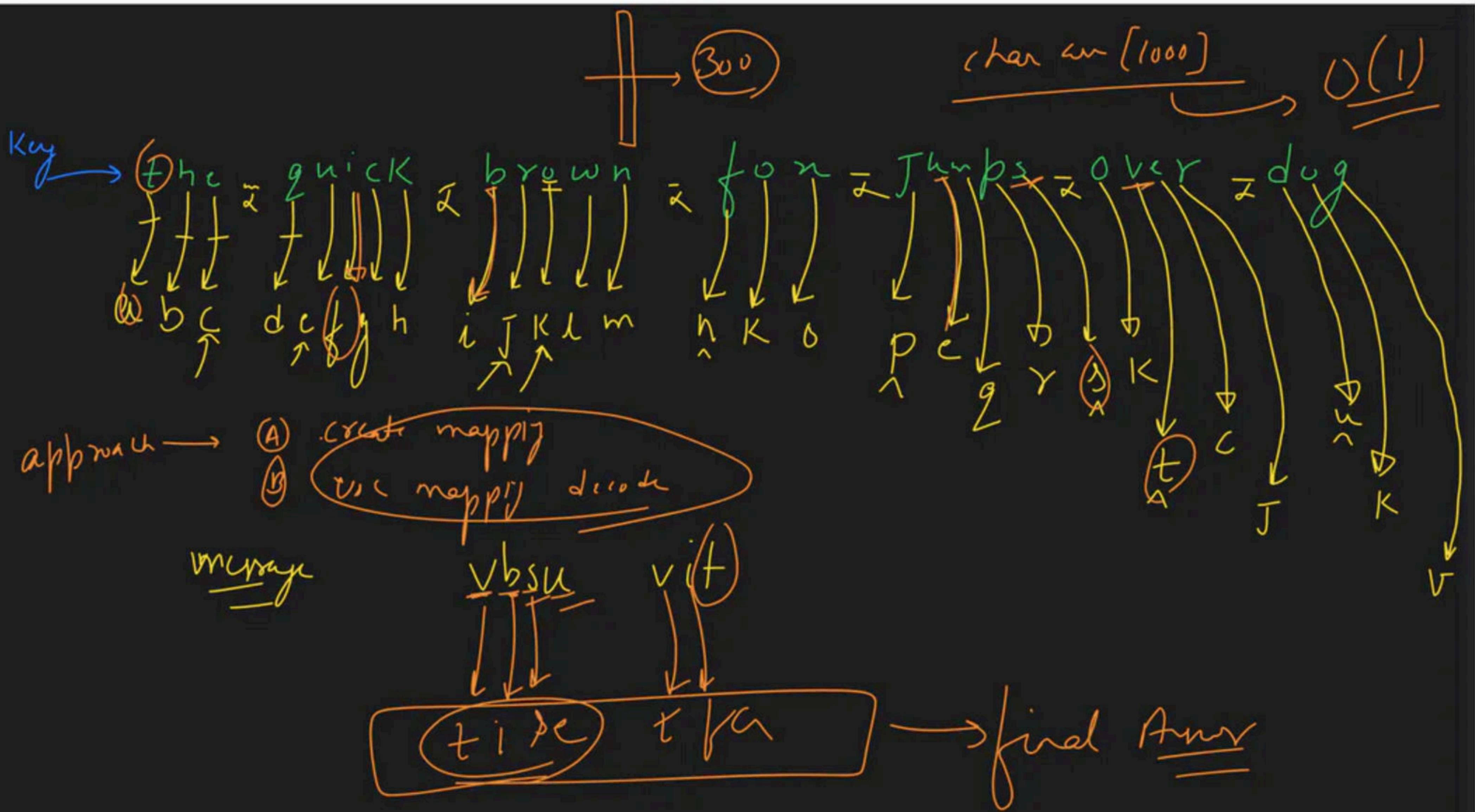


Key →



$t \rightarrow a$
$h \rightarrow b$
$c \rightarrow c$
$d \rightarrow d$
$o \rightarrow e$
$g \rightarrow f$

S.T



Key:

t	h	c	'	g	u	i	c	k	'	b	v	o	w	n	'	f	o	n	'	j	u	m	p	s	'	'	

start

'a'

mapping

0	0	0	0	0	-	-	-	-	-	\	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

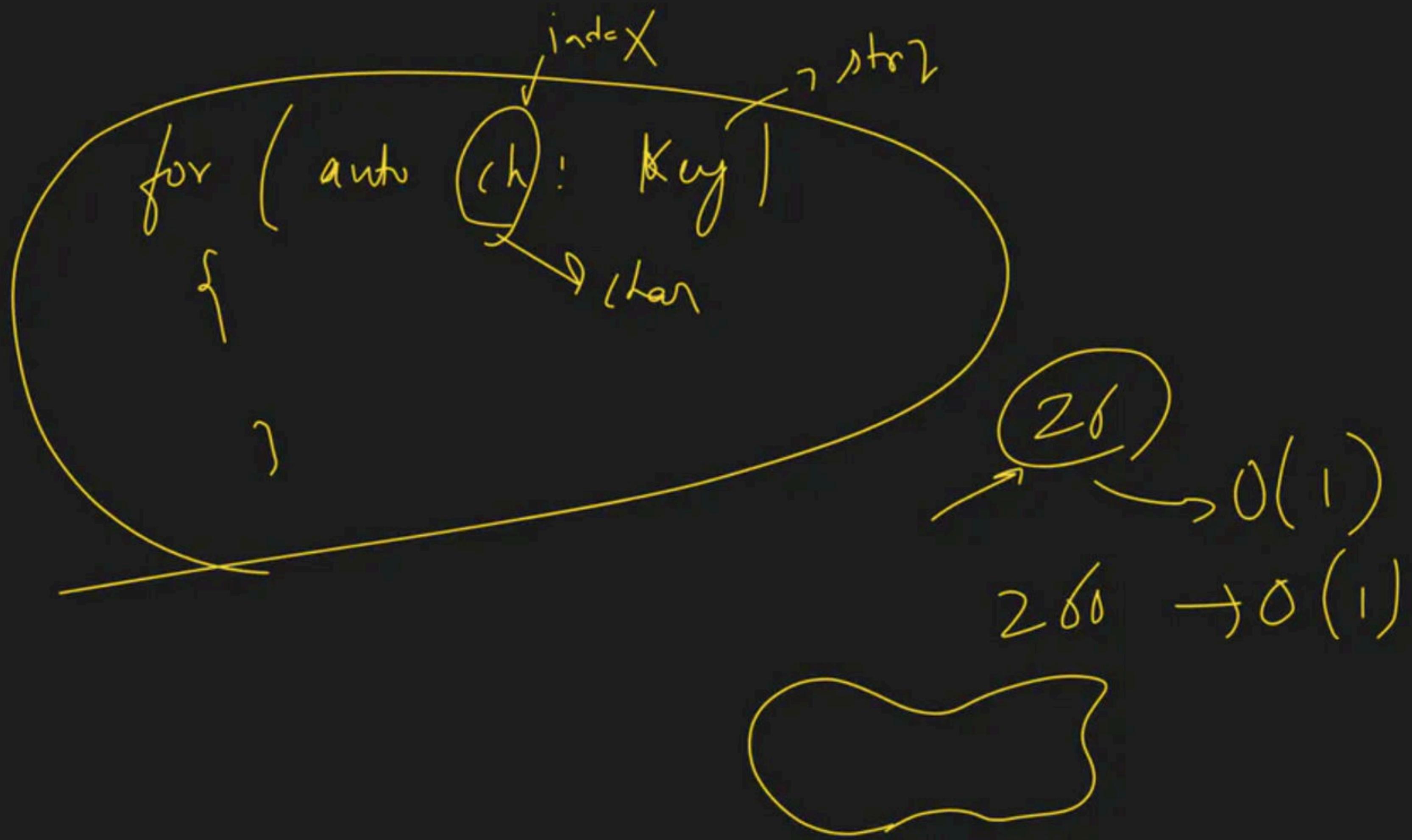
create mapping

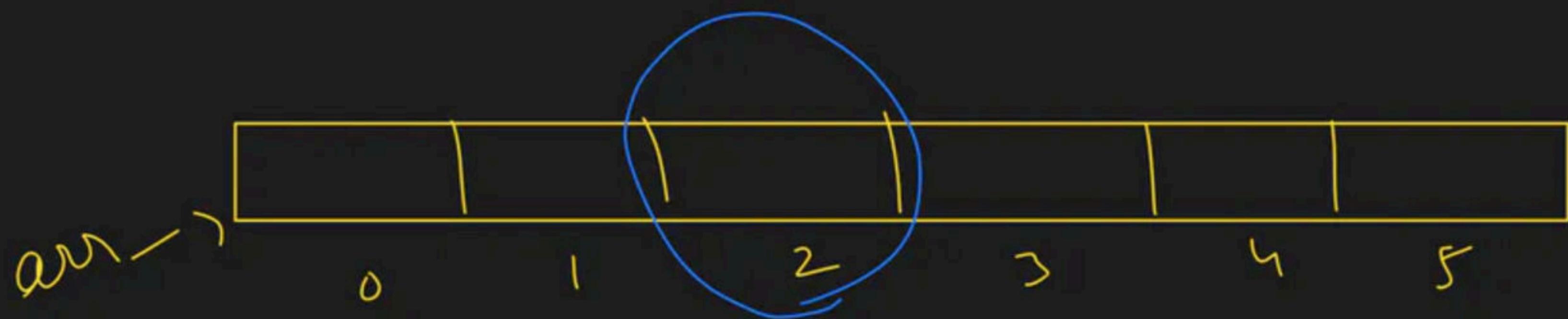
char start = 'a'

char mapping[300] = {0}

for (auto ch : key)

if (not isalpha(ch) == 0)
mapping[ch] = start
start

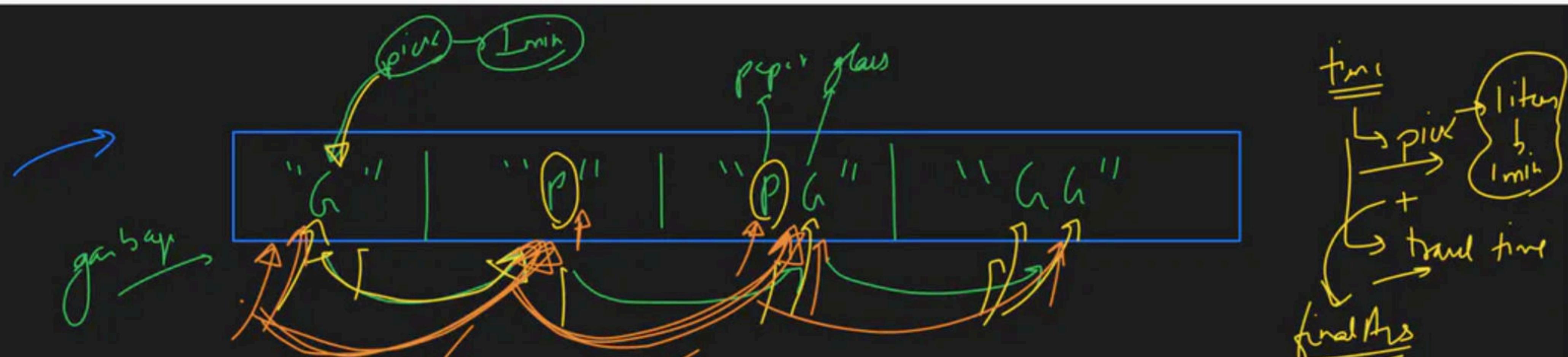




arr [2]

ch = ~~a~~

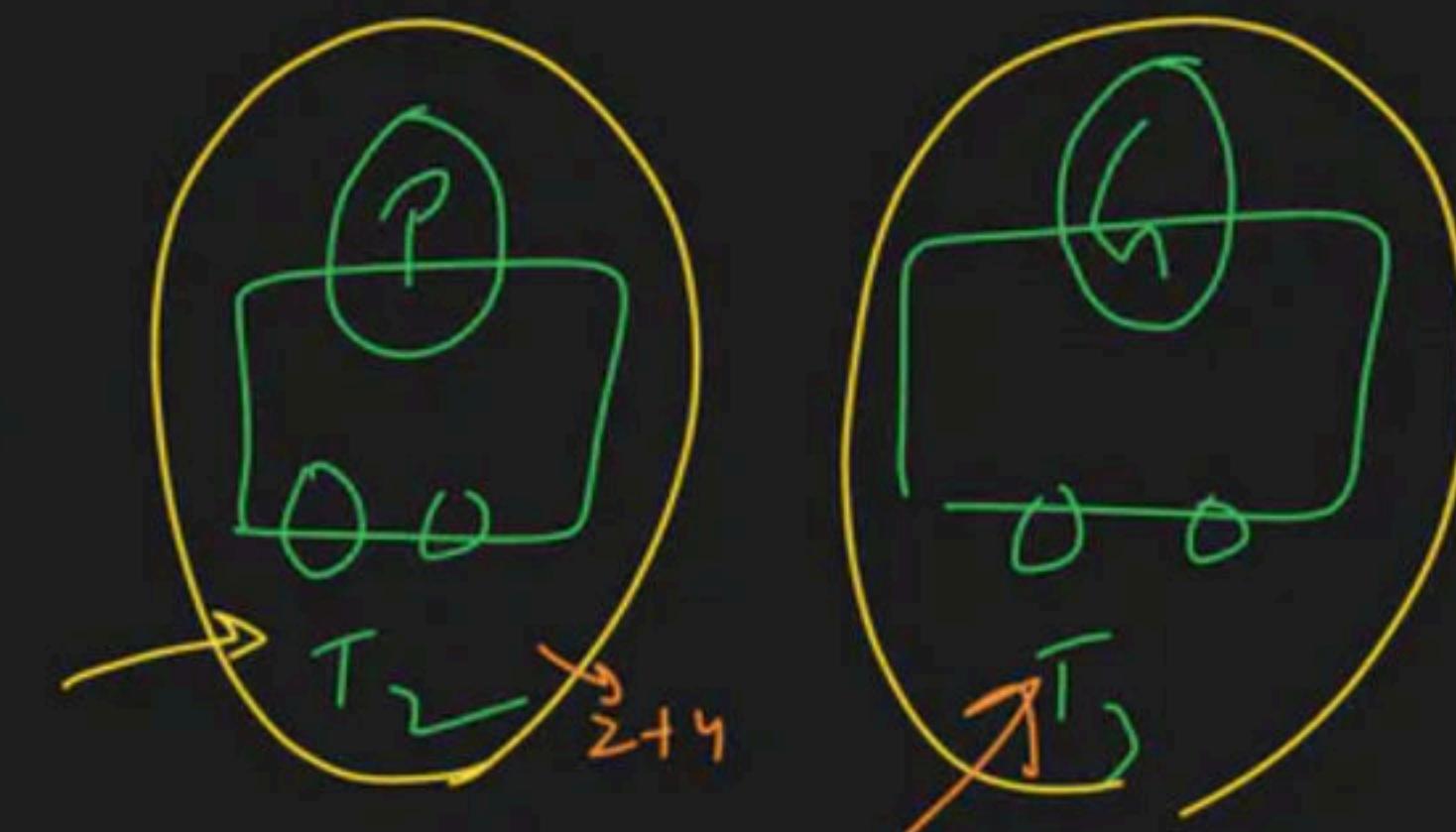
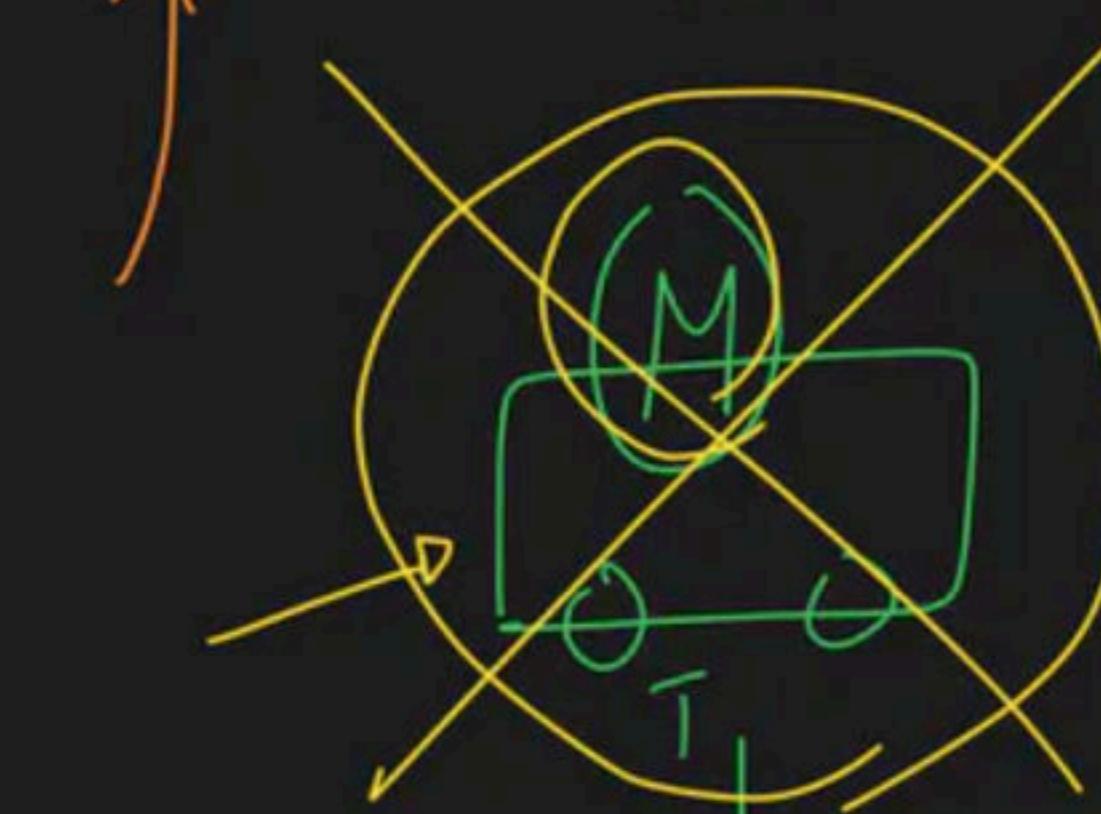
mappy [ch]
g7



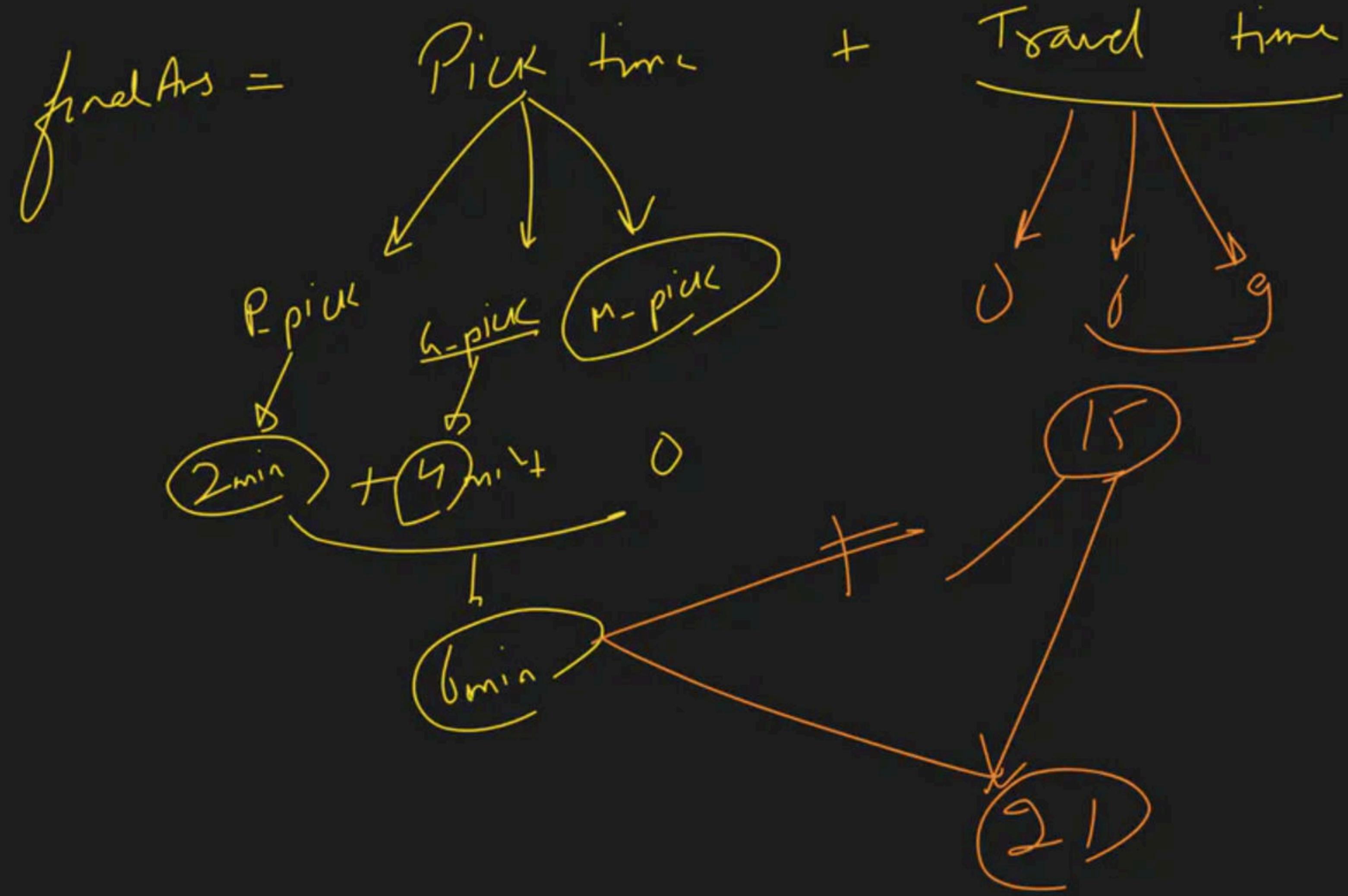
~~travel~~ $\overline{\text{travel}(i)}$

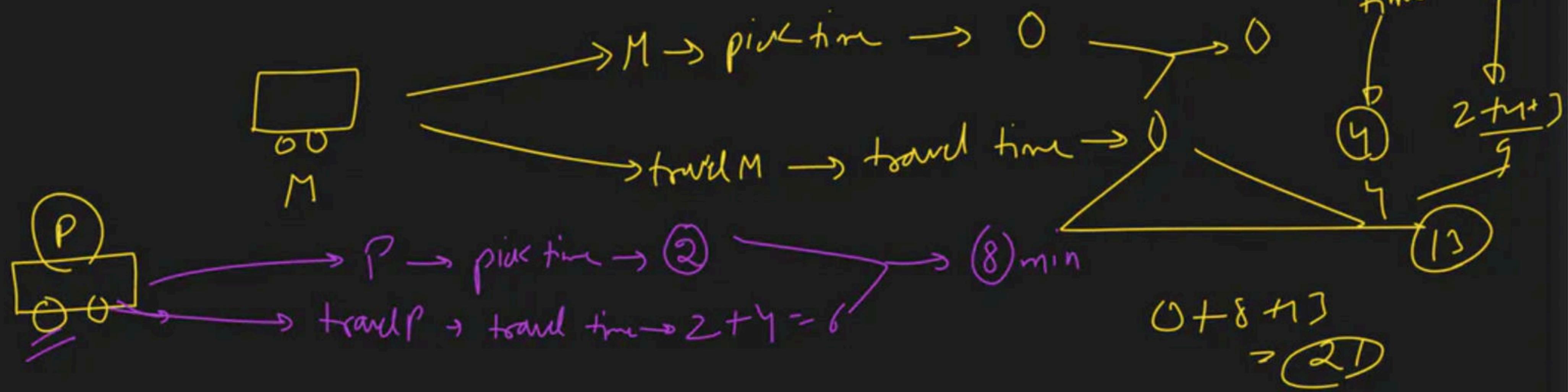
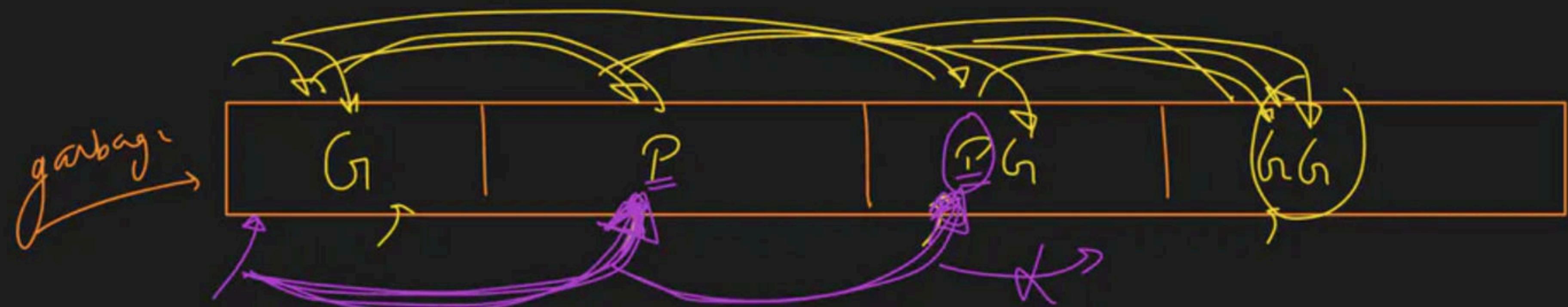
$\overline{\frac{1}{\text{min}}}$

$\overline{\text{garby}(i) \rightarrow \text{garby}(i+1)}$



find min no of
minutes to
pick all
garbage





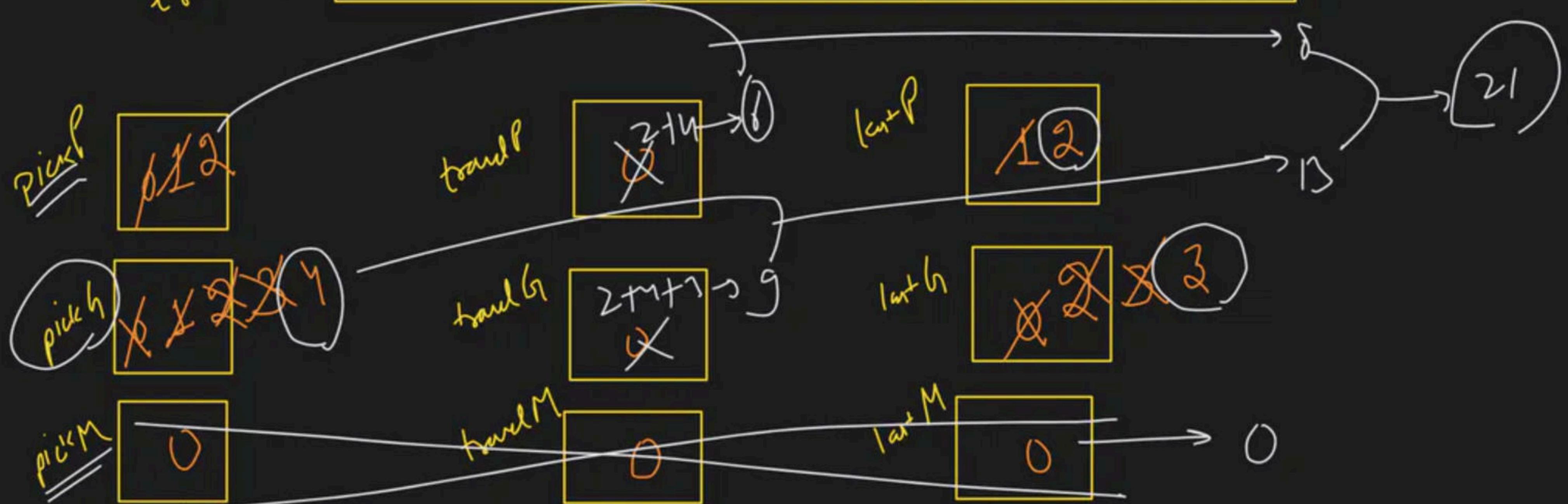
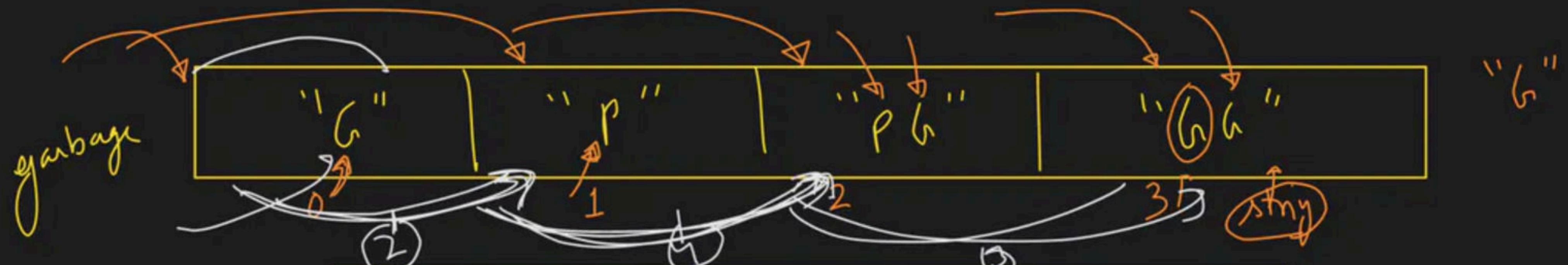


Final Ans =

(Pick time + travel time) P truck

+ (pick time + travel time) h truck

+ (pick time + travel time) M truck

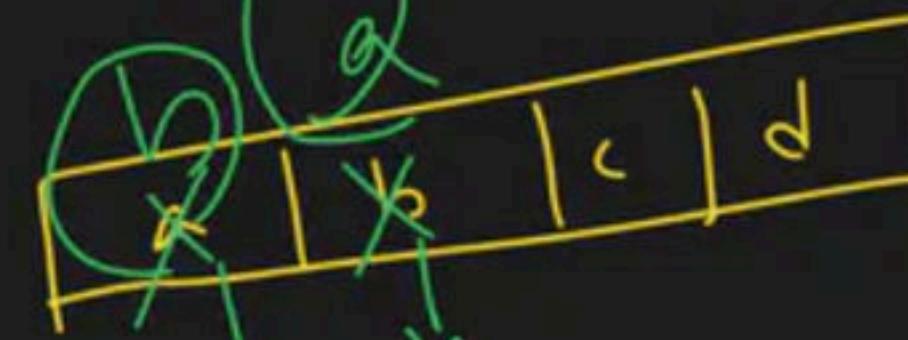


$\text{order} = \langle$



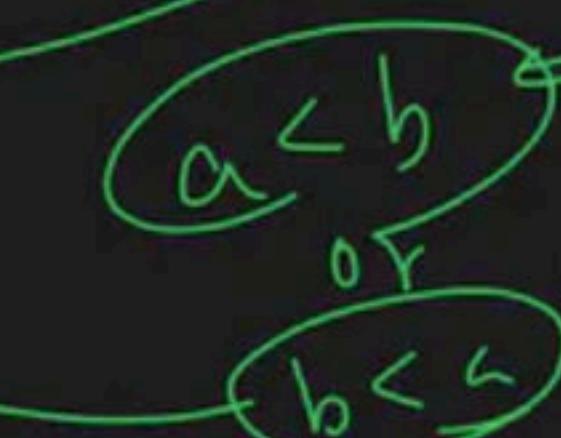
$s = \langle a b c d \rangle$

sort



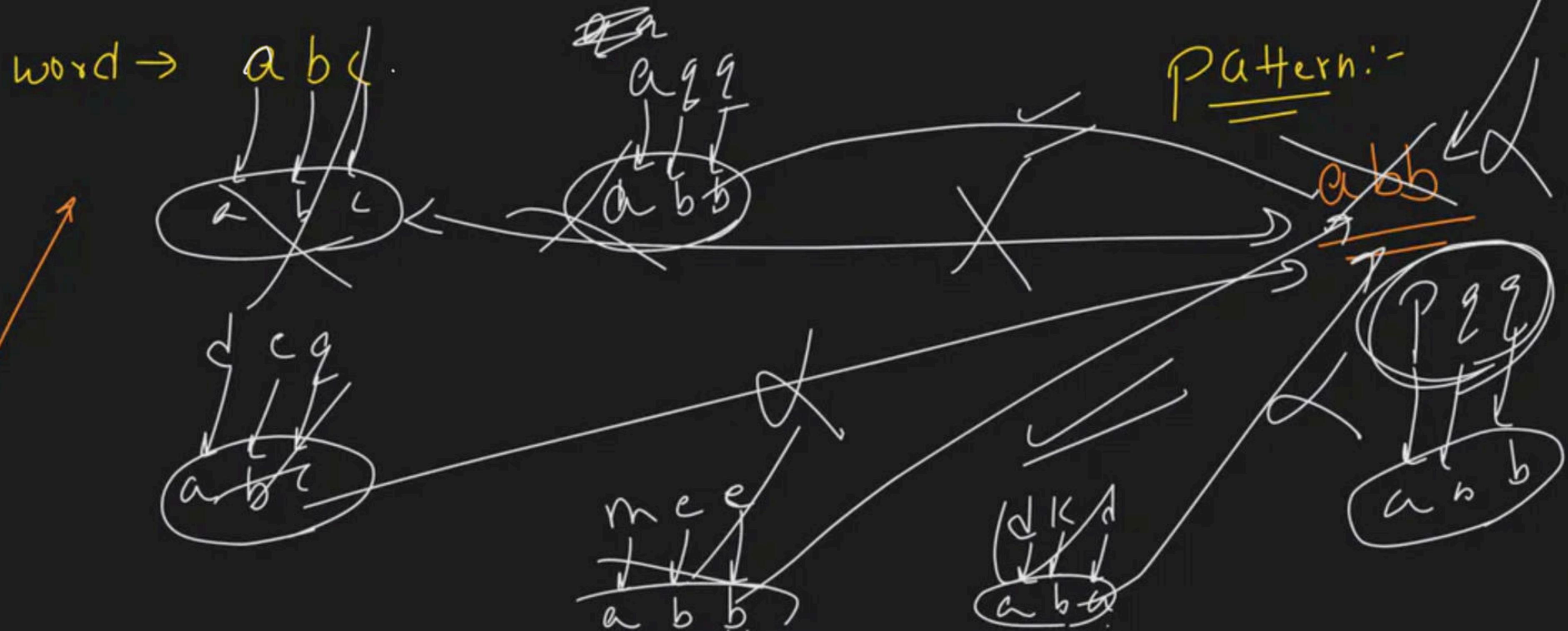
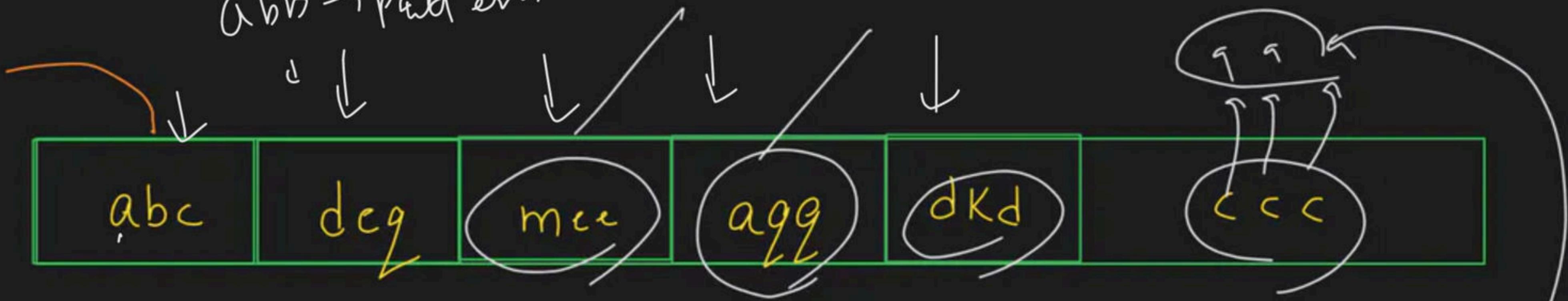
$(s \cdot \text{begin})$

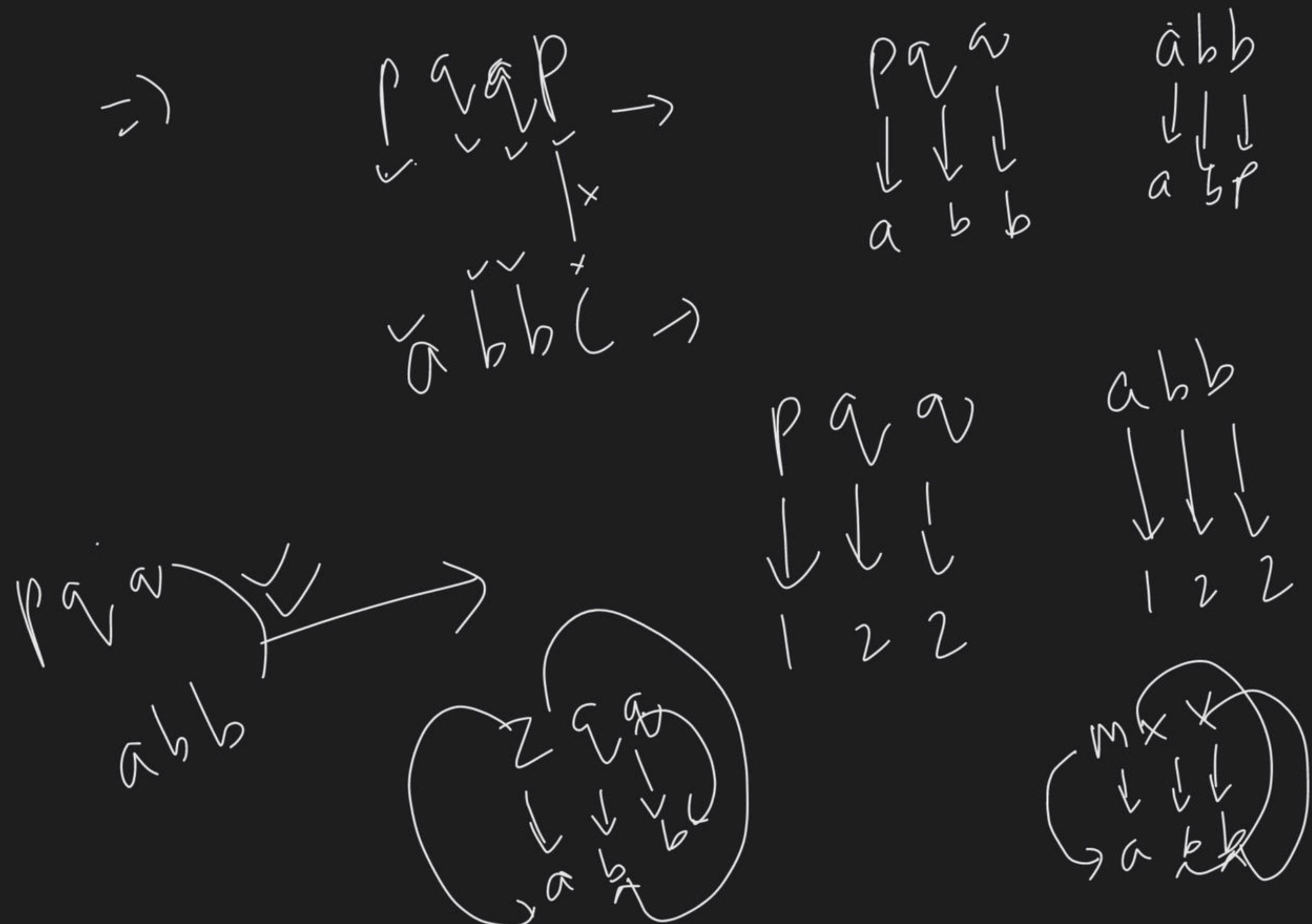
$(s \cdot \text{end})$

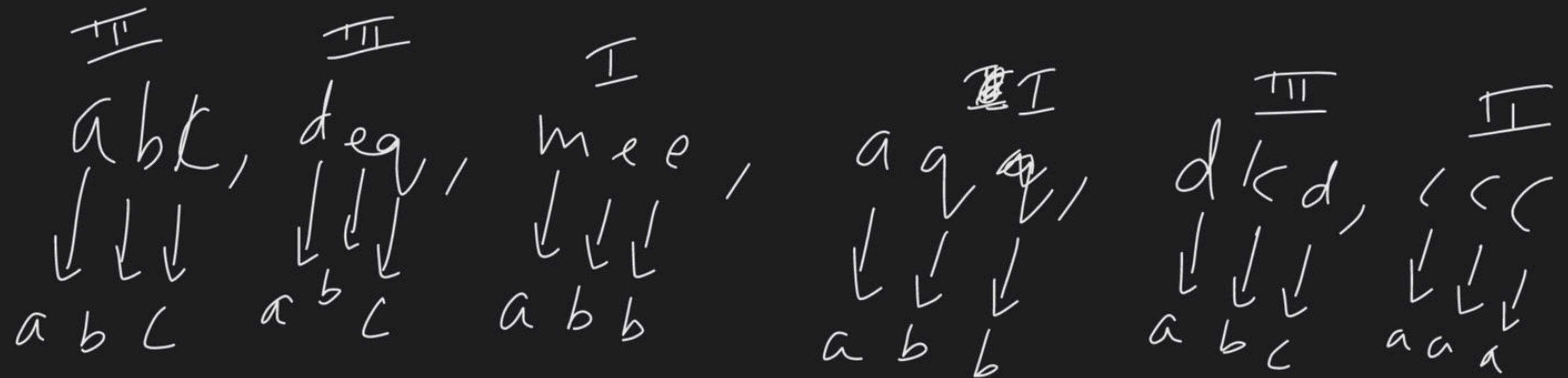


γ_{min}

$abb \rightarrow$ part even





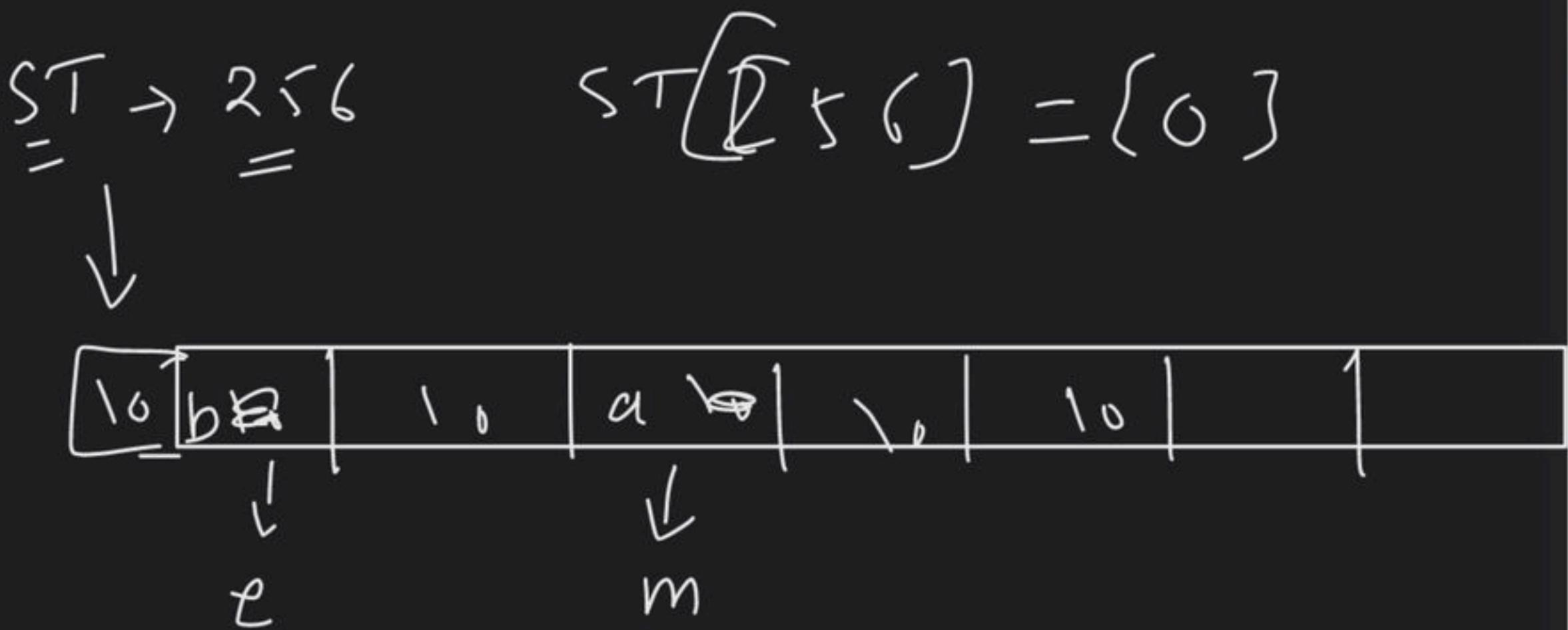


P \Rightarrow abc
 III

Pattern:
 aa&a II

Pattern
 abc I
 ↴ ↴
 a b b

\Rightarrow
 m e e
 \Rightarrow

$\text{ST}[m] = 'a'$ \rightarrow char. \hookrightarrow $\text{char} = 'a'$
 $\text{ST}[e] = 'b'$ \rightarrow $'b'$
 $\text{ST}[e] = 'b'$


①

Quantum nail horse



① \neq

aplication
Bouls =



Graph works

② Clean work + follow



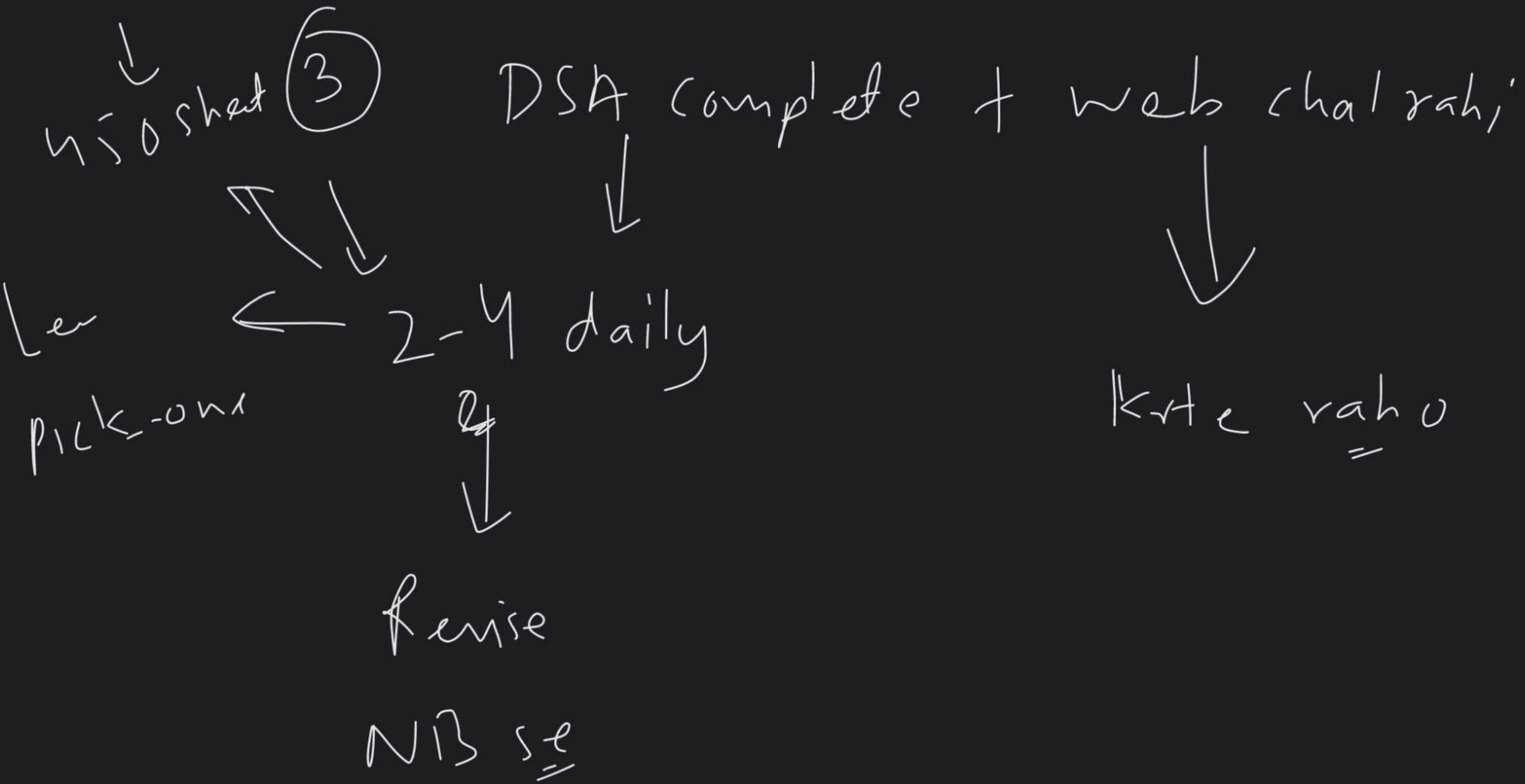
100% explanation to M

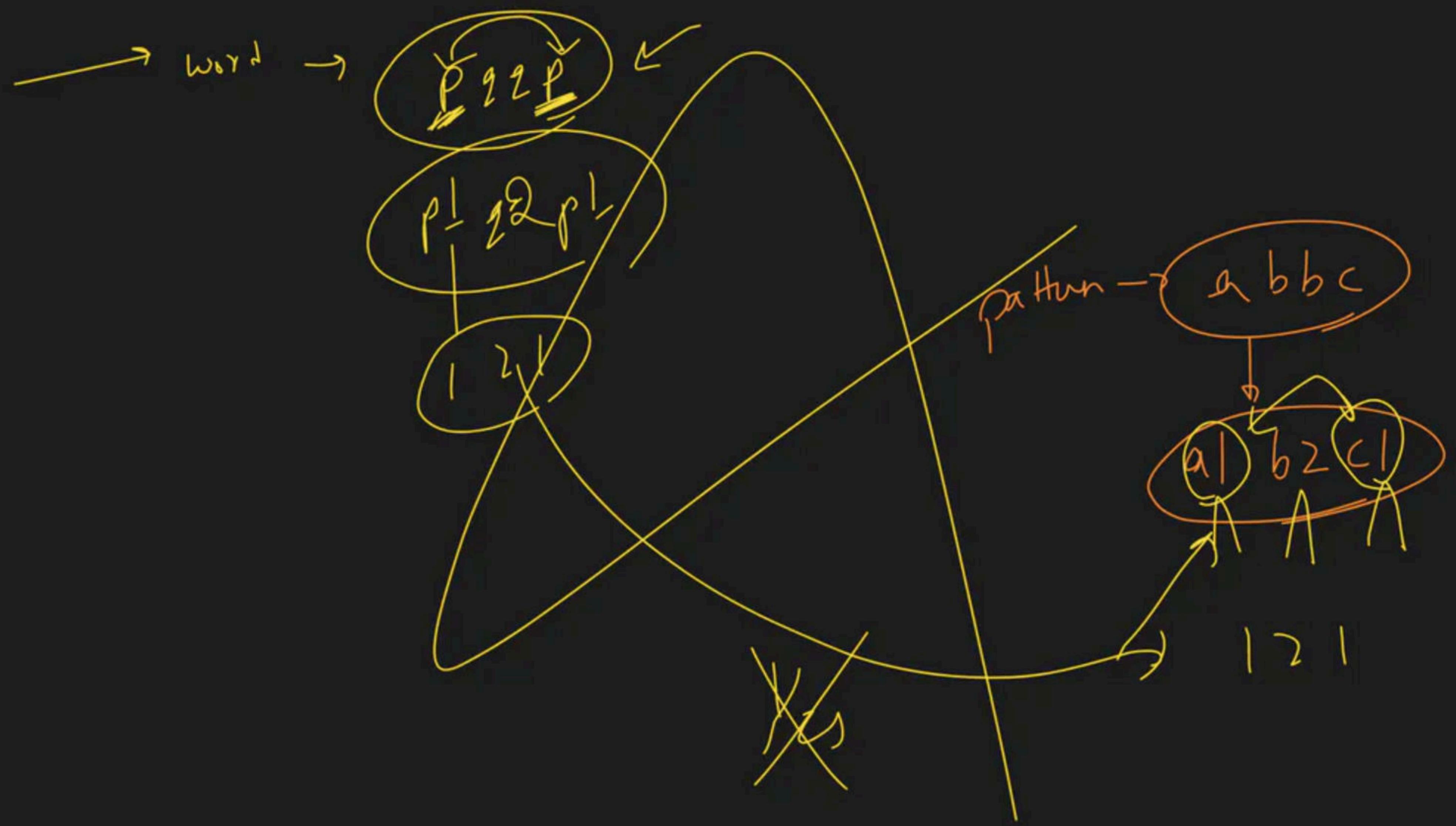
② DSA + Web \rightarrow App

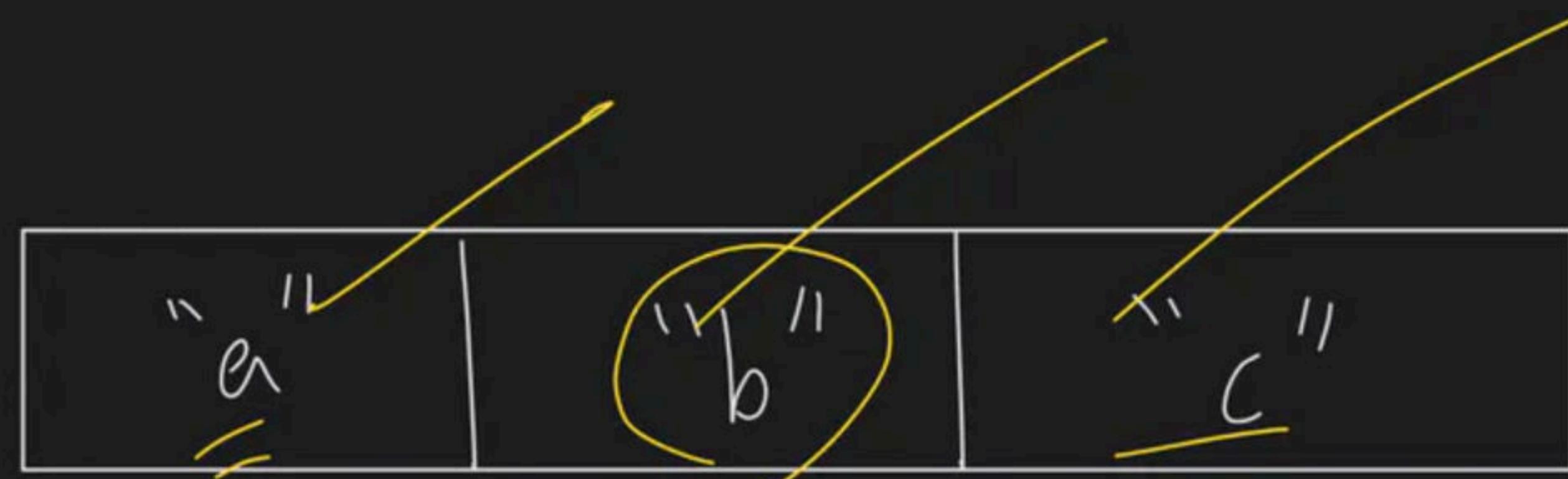
① DSA X Web X

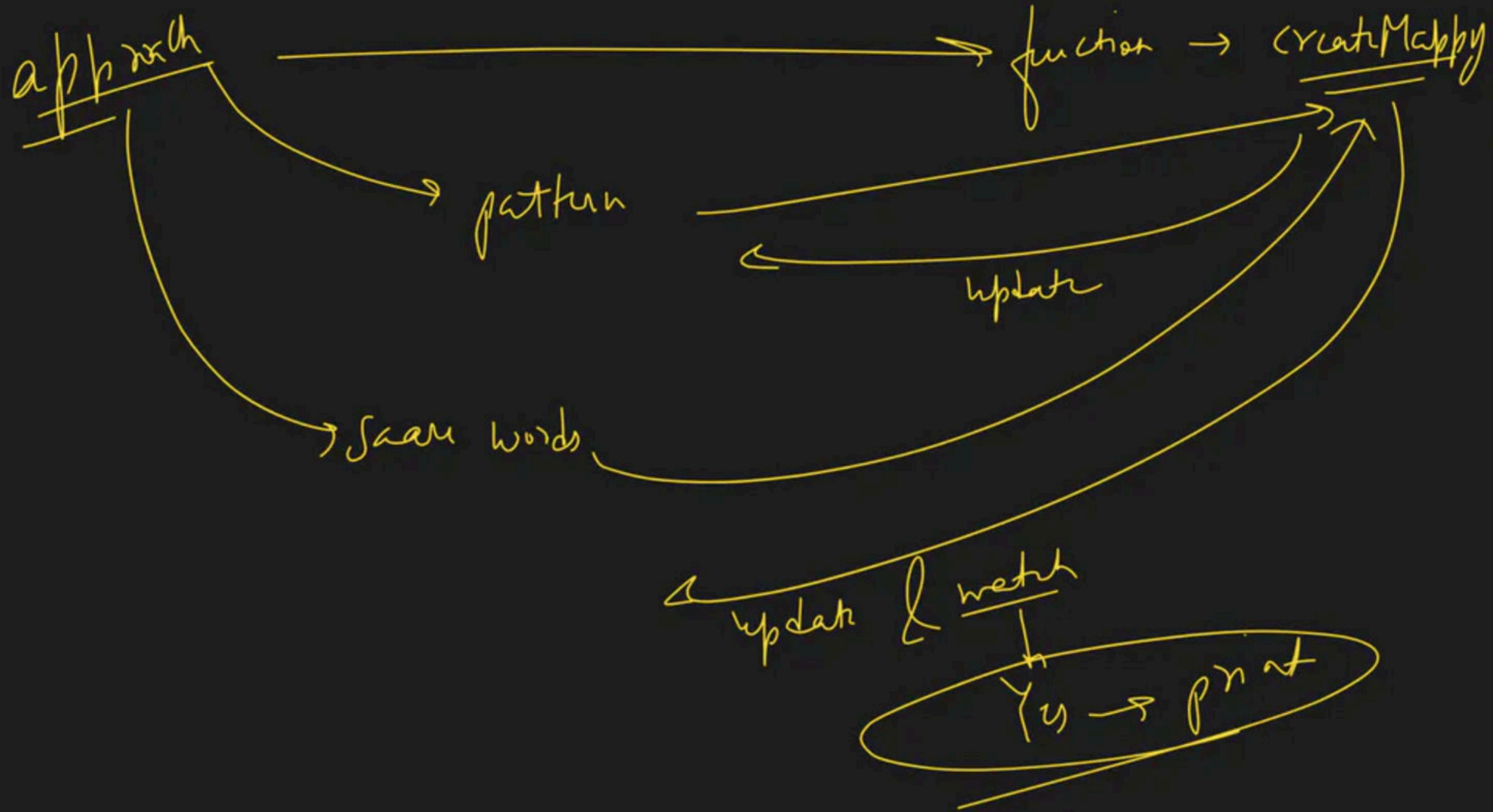
\hookrightarrow ① 3 Months Srf DSA
 $=$

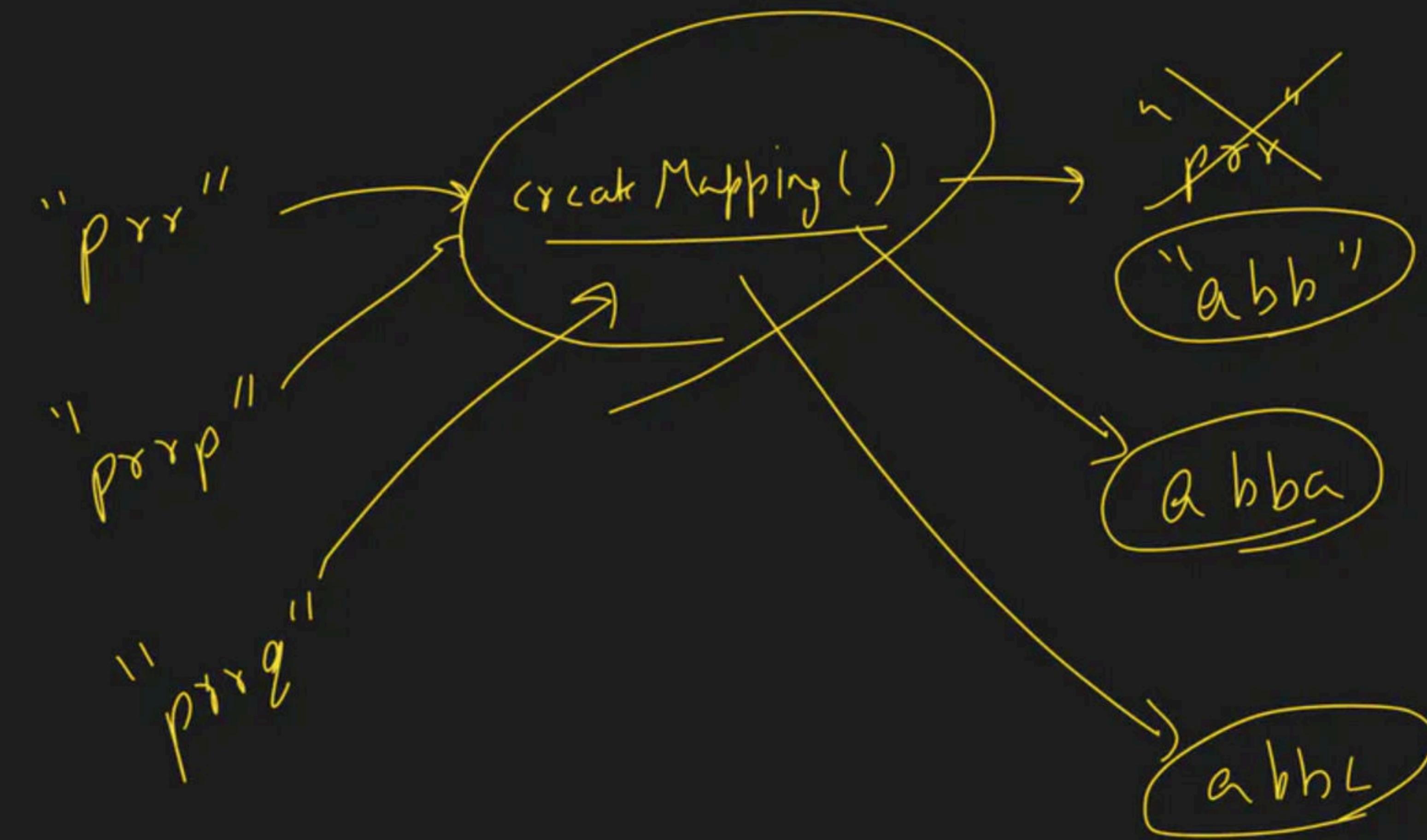
② 4 months X DSA + Web
4 days \downarrow
 \downarrow 3 web =

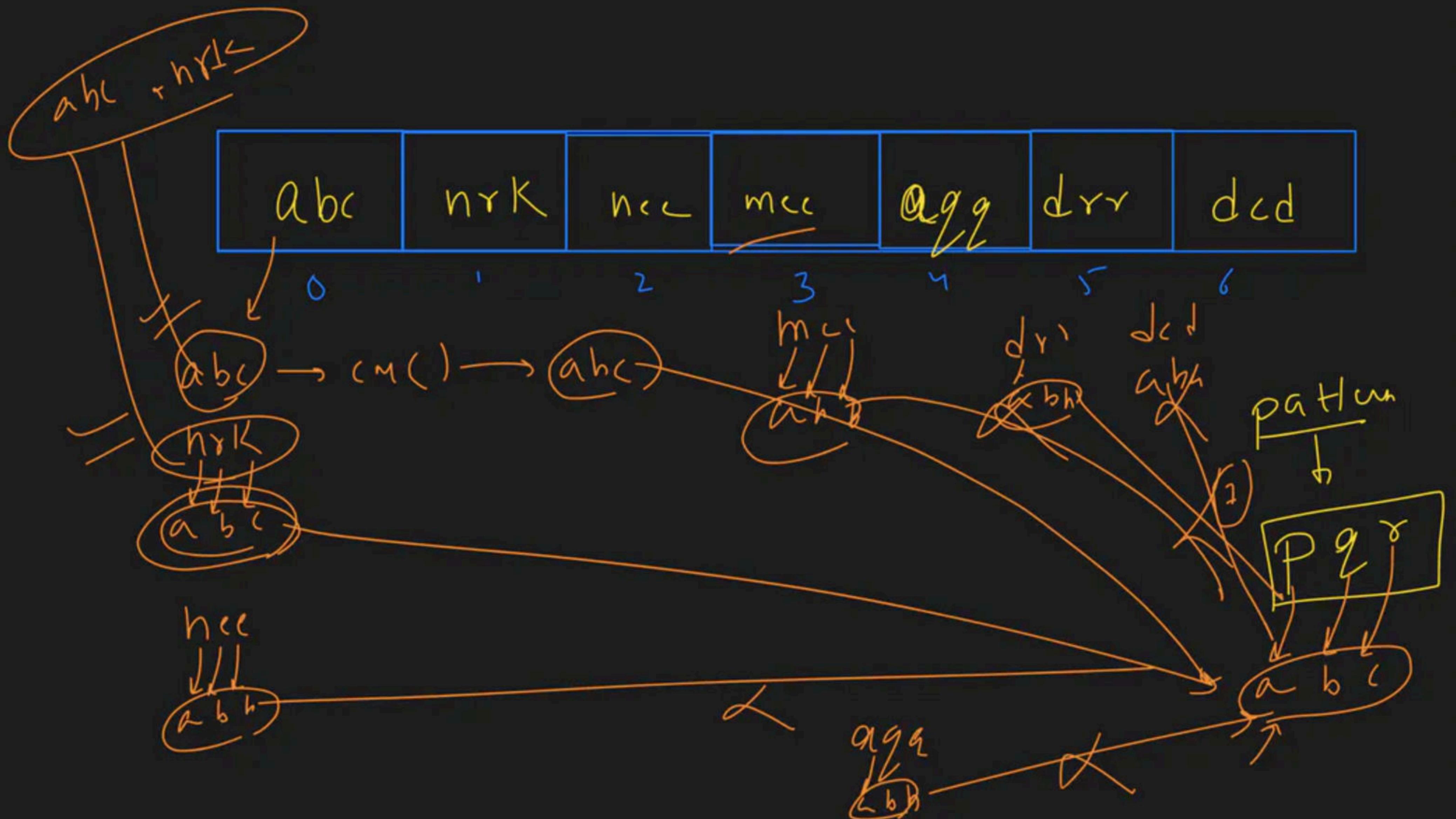












Y sul

Tic | sic

et plair

BONUS

Vander

















