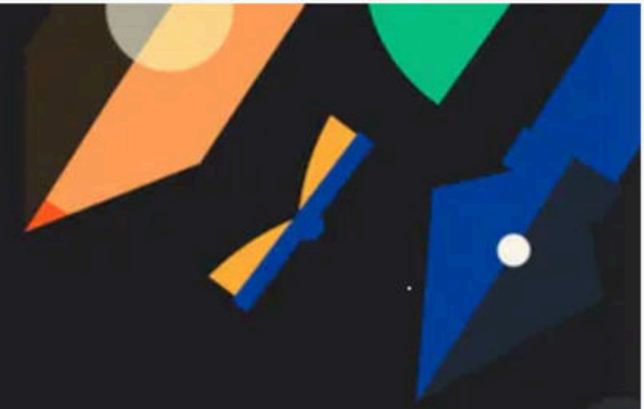


Character Arrays & String

Foundation Course on Data Structures & Algorithm - Part I

Love Babbar • Lesson 14 • May 12, 2022

Char arrays & Strings

array → H/w

video

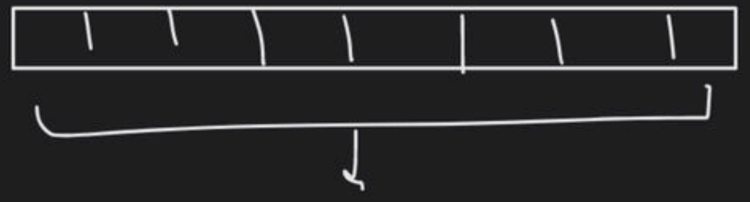
Sunday

Shi

→ char → ?

DT

1 Byte = 8 bits



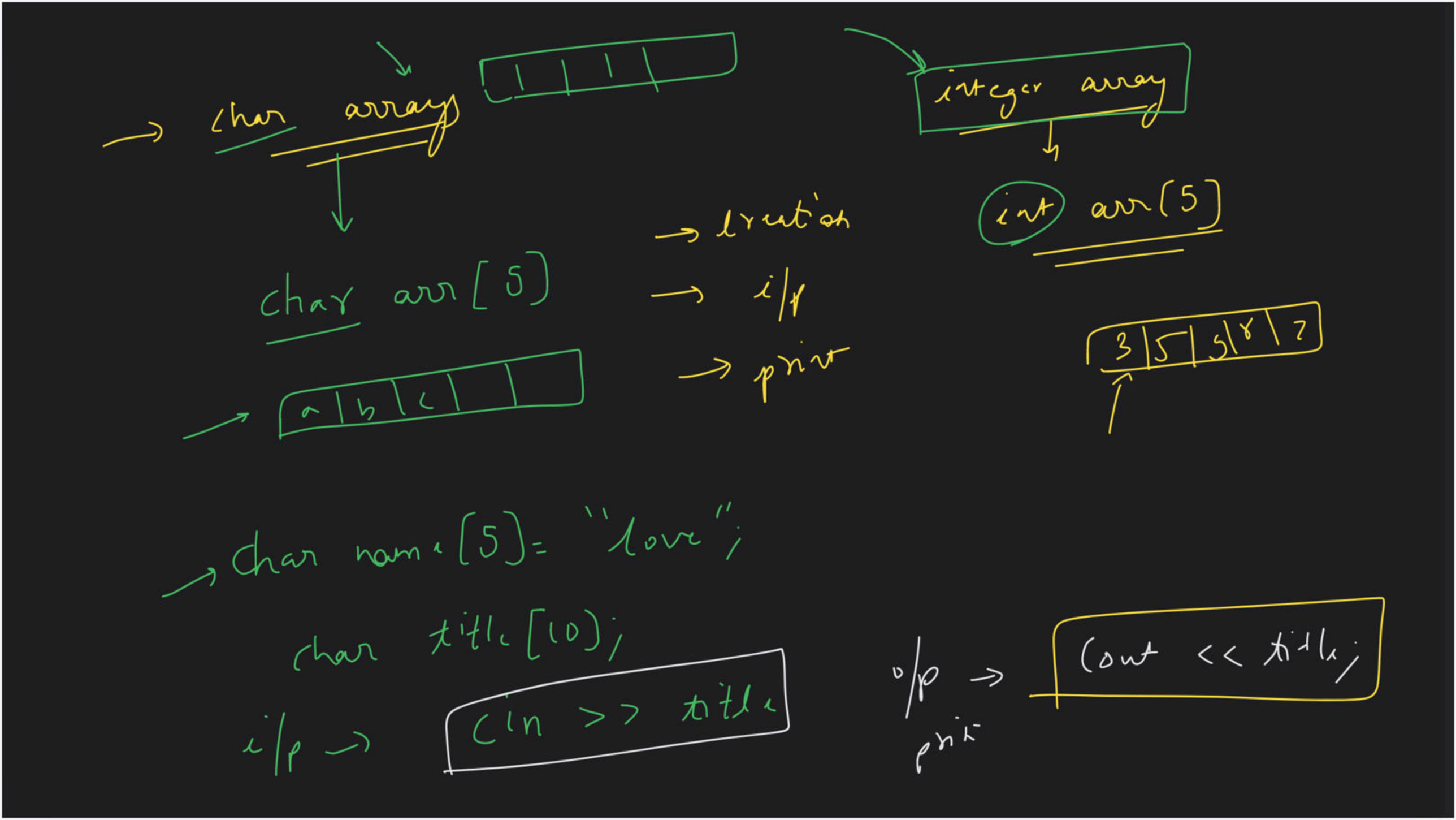
character

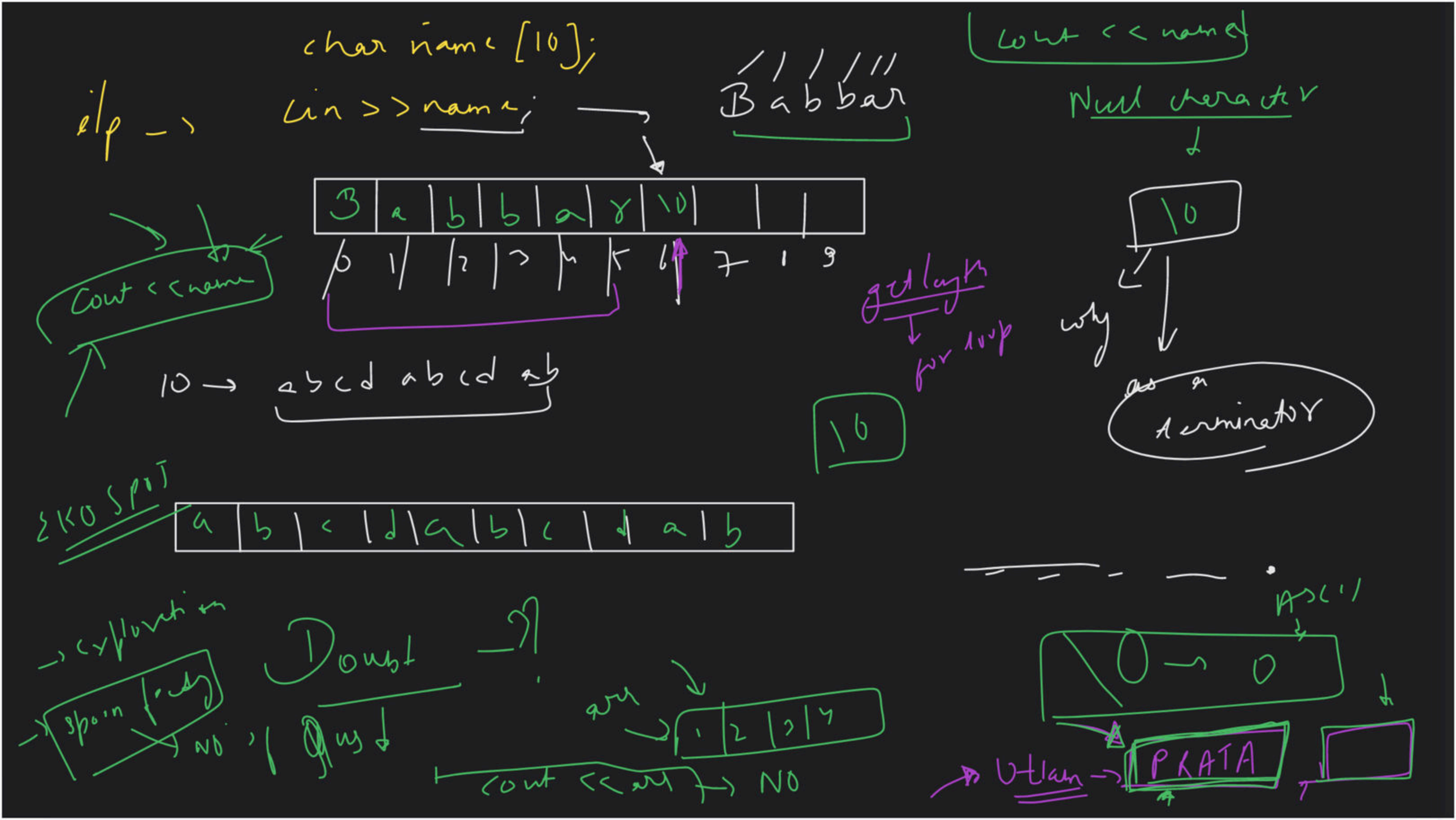
char → ASCII value

ASCII table

'A' → 65

'a' → '97'


char array
integer array
int arr[5]
char arr[5]
→ location
→ i/p
→ print
a b c
3 5 5 8 2
char name[5] = "love";
char title[10];
i/p → cin >> title;
o/p → print
cout << title;


char name[10];
i/p ->
cin >> name;
Babbar
cout << name
Null character
3 a b b a r \0
cout << name
10 -> abcd abcd ab
getlength
for loop
while
10
Terminator
EKO SPOT
a b c d a b c d a b
-> explanation
Doubt
spoon
yes
NO
arr
1 2 3 4
cout << arr -> NO
ASCII
0 -> 0
PRATA
Utkarsh

→ possible sol (mid) (Brute force)

int time = mid

t = t − K * arr(i)

target

if

toLeft int t = time

while (t > 0

{ t = t − K * arr(i)

App
- KHUD
- Init
- Editorial
- Discord →

Love Babbar

R
- DR
- DR
- DR

→ CK's POT

→ ROT1 → Min

Video

-> char -> operation

i/p

o/p

1 D A

int array

40 B

10 int

mem

40 Byte

char array

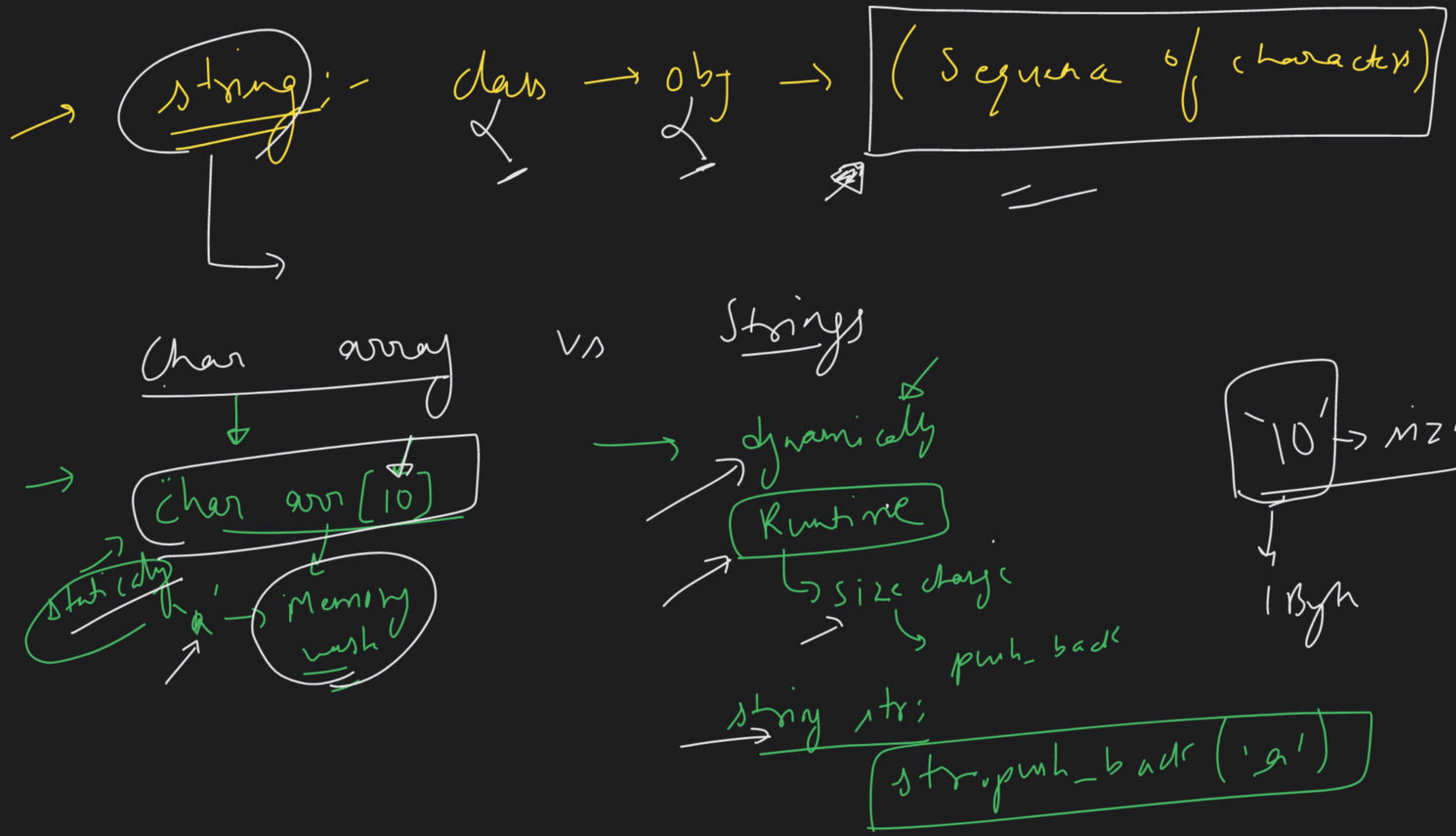
10 B

10

10 char

mem

10 Byte


string :-
class
obj
(Sequence of characters)
Char array
vs
Strings
char arr [10]
Static city
'a'
Memory wastage
dynamically
Runtime
size change
push_back
'10' → size
1 Byte
string str;
str.push_back ('a')

CA

→ fast

→ less
pindefined
func

→ char → getlength()

String

slow incomparison to CA → why?

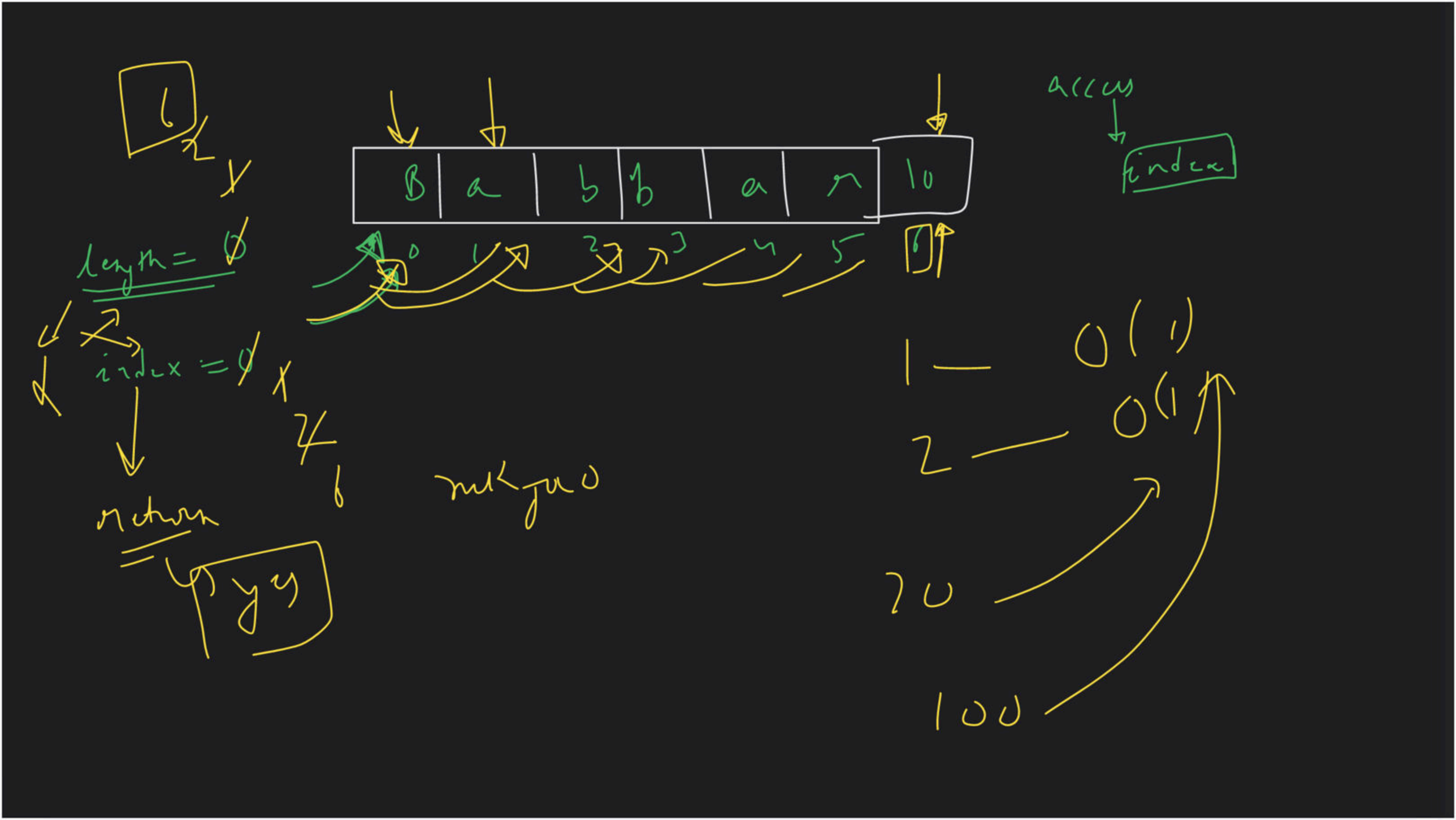
more

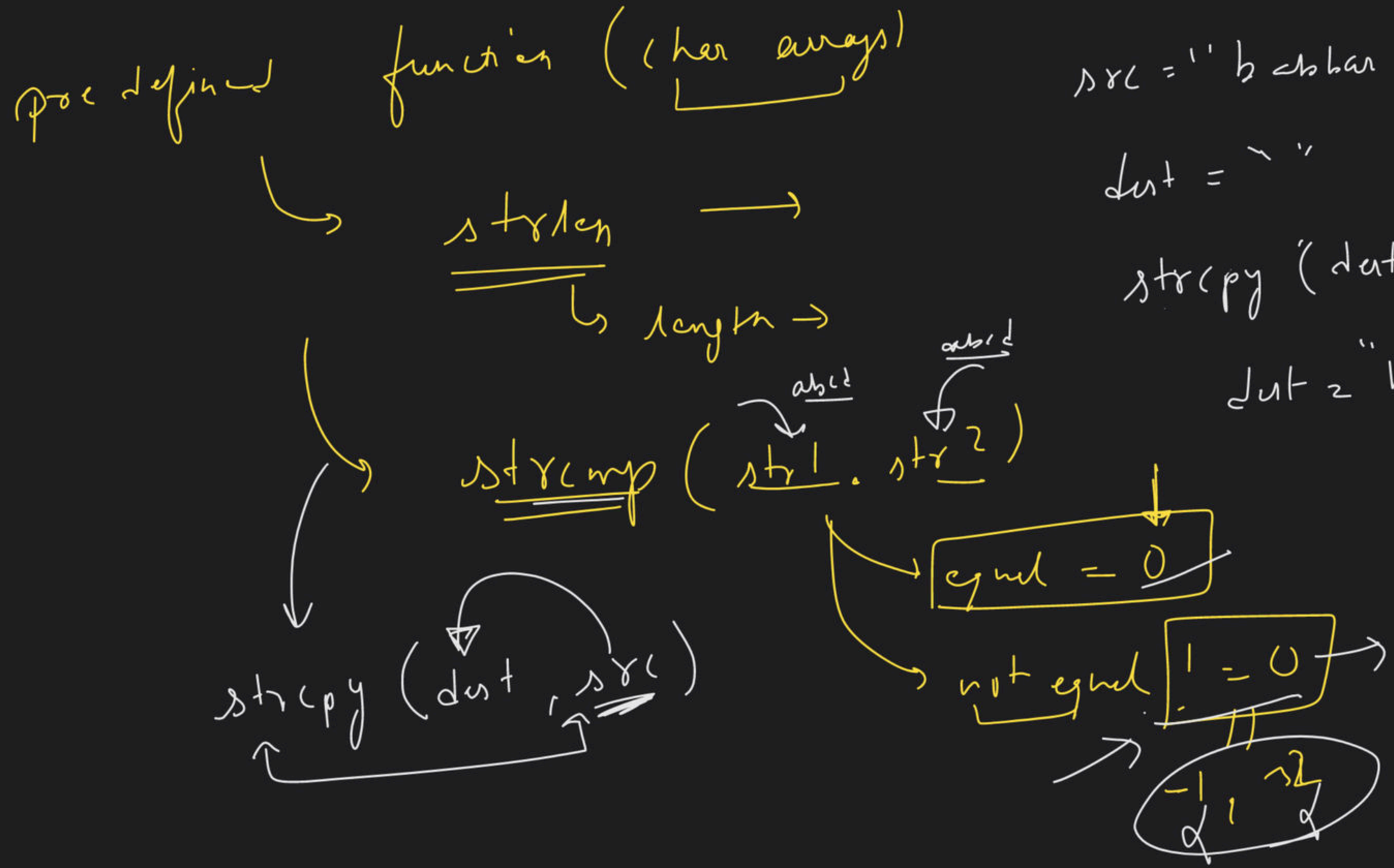
push_back

→ code

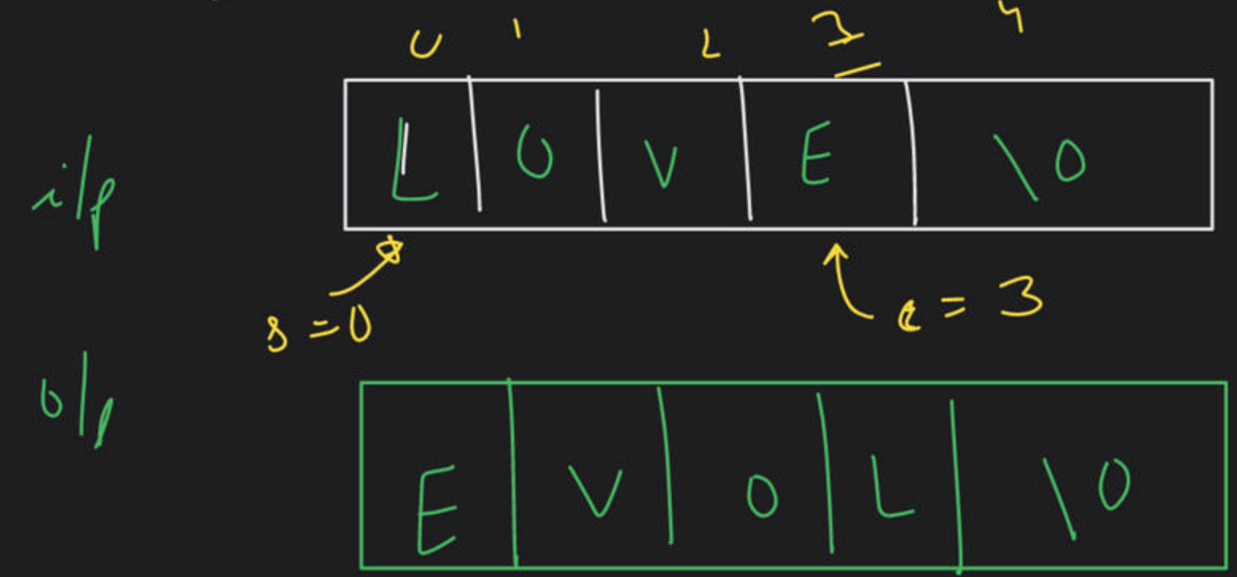
13 LP
23 LPA

13 May
1 cm
2 min
Life


B
a
b
b
a
n
\0
access
index
length = 0
index = 0
return
yy
O(1)
O(1)
70
100


pre defined function (char arrays)
strlen
length
strcmp (str1 , str2)
str1
str2
equal = 0
not equal
1 = 0
-1
1
32
strcpy (dest , src)
src = "barbar"
dest = ""
strcpy (dest, src)
dest = "barbar"

Reverse a C.A



while(s <= e)

length = strlen(name);

s = 0

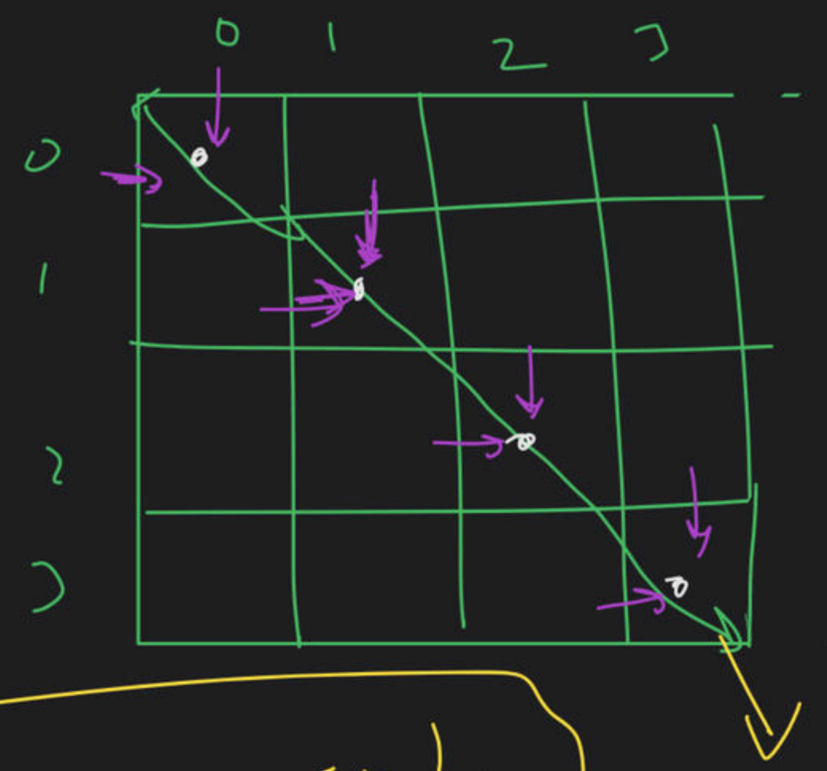
i <= length - 1

e = strlen(name) - 1

Char

stolen()

size



int arr

size

fn (arr, n)

i = 0 j = 0

i = 1 j = 1

i = 2 j = 2

i = 3 j = 3

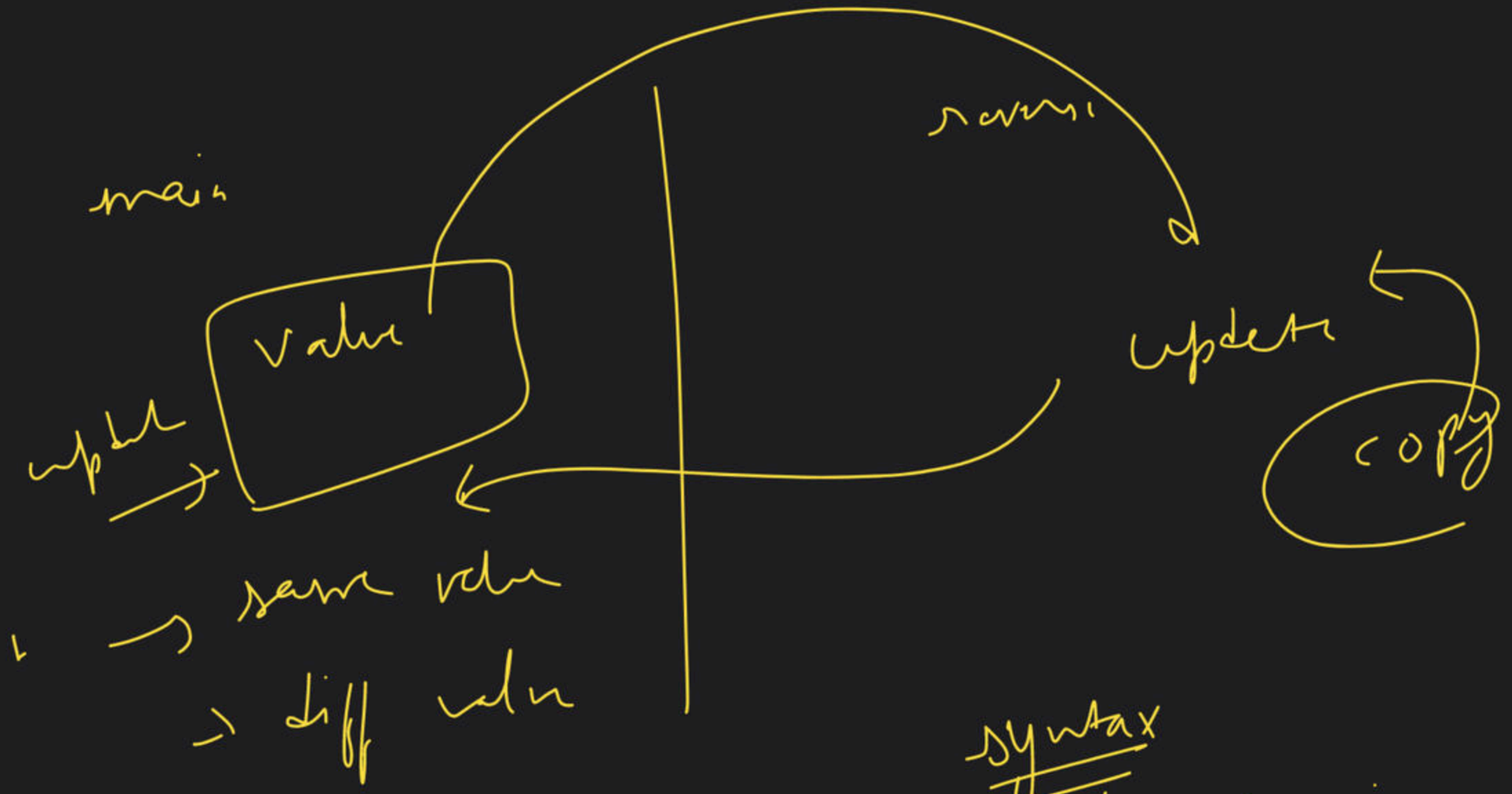
Total

```
for( int i = 0 ; i<n)
{
sum = sum + arr[i][i]
}
```

$O(n^2)$

$O(n)$

Yes or No



syntax

↳ practice

→ Reverse

string

string str;
cin >> str;

↳ space
tab
newline/enter

→ Delimiter

Discord
↳ Discuss

getline(cin, str);

Custom
delimiter

Bugambo

space

Kumar Jain

space

→ babbar

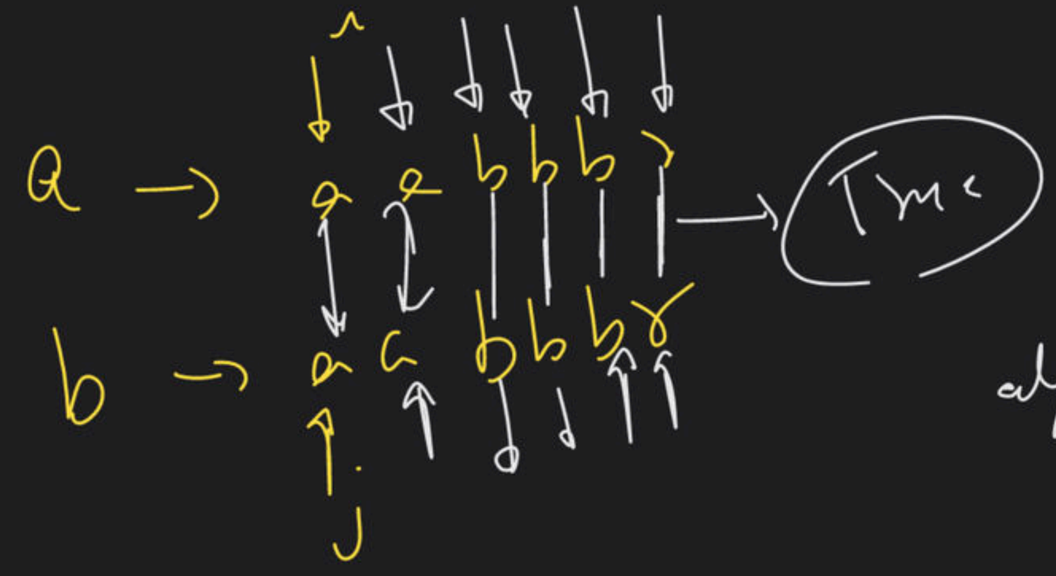
babbar


0
1
B
a
arr
arr[1]
a

→ string a = "babbar" ;

string b = "brabbac" ;

whether b is anagram of a or not ?



Approach

Diff

↓

16 ms

&

compare

solution

anagram

↳ same characters

↳ ordering different

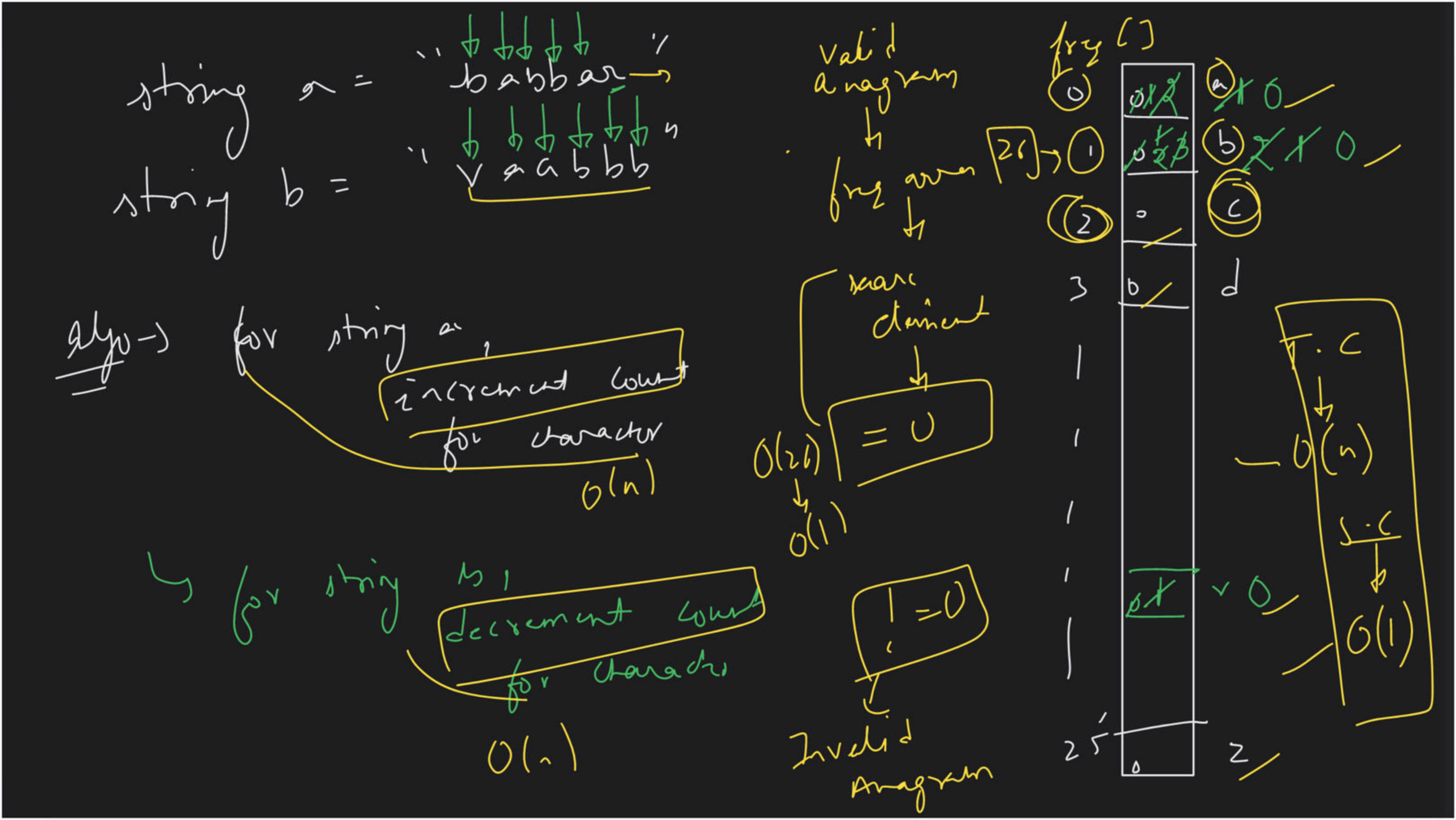
algo

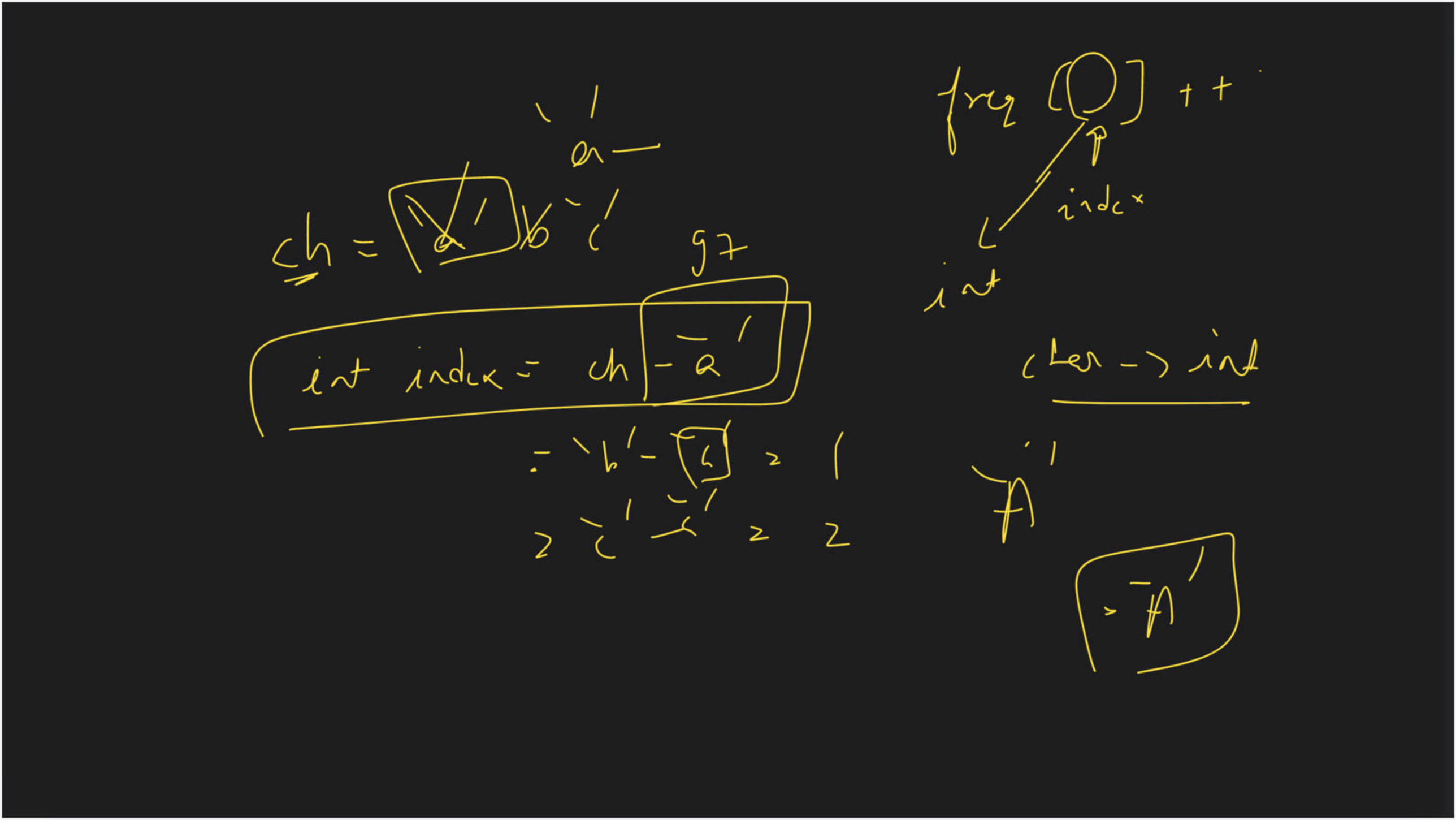
↳ sort a → nlogn

↳ sort b → nlogn

↳ compare → O(n)

→ O(nlogn)

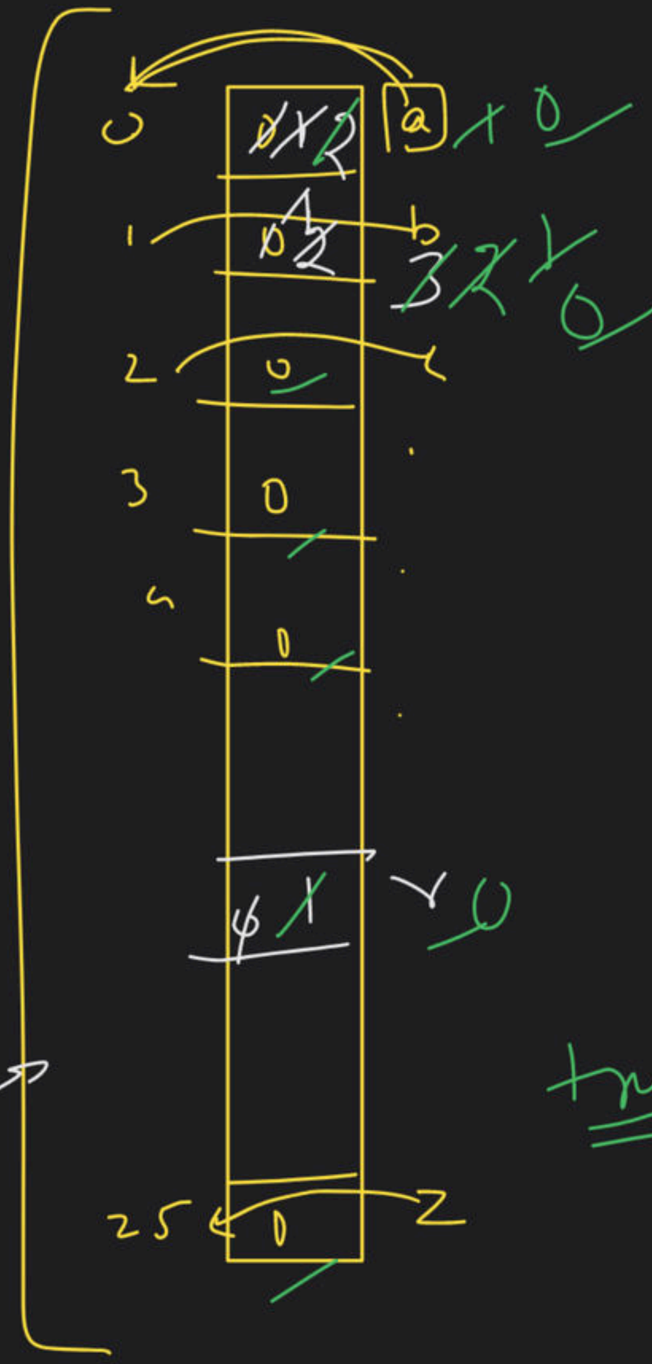

string a = babbar
string b = aabbb
Algo:- for string a,
increment count for character
O(n)
for string b,
decrement count for character
O(n)
Valid anagram
freq array
search element
= 0
O(1)
O(1)
! = 0
Invalid Anagram
freq []
T.C
O(n)
S.C
O(1)


freq [ ] ++
index
int
ch = 'a'
'b'
'c'
97
int index = ch - 'a';
= 'b' - 'a' = 1
2 'c' - 'a' = 2
(char -> int)
'A'
= 'A'

Doubt

a = "babbab"

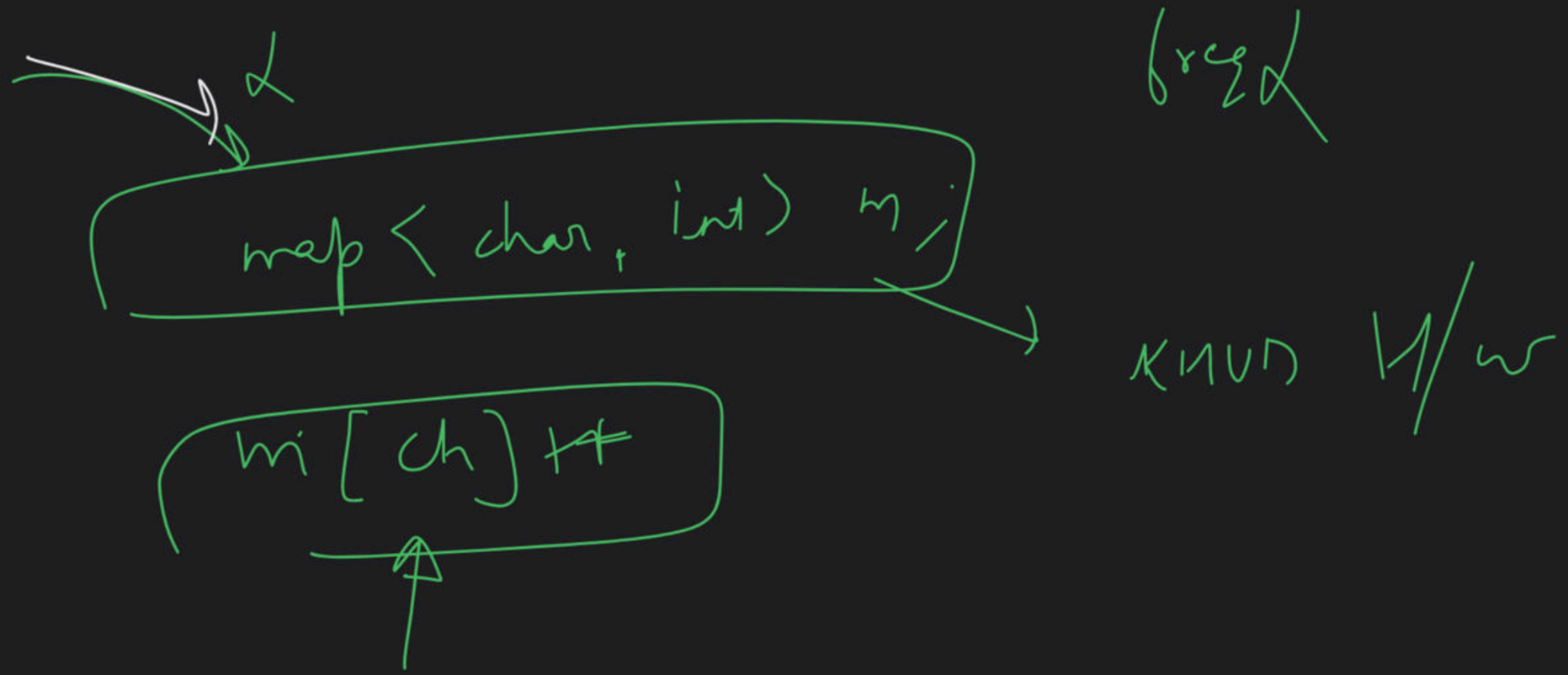
b = "aabbbb"

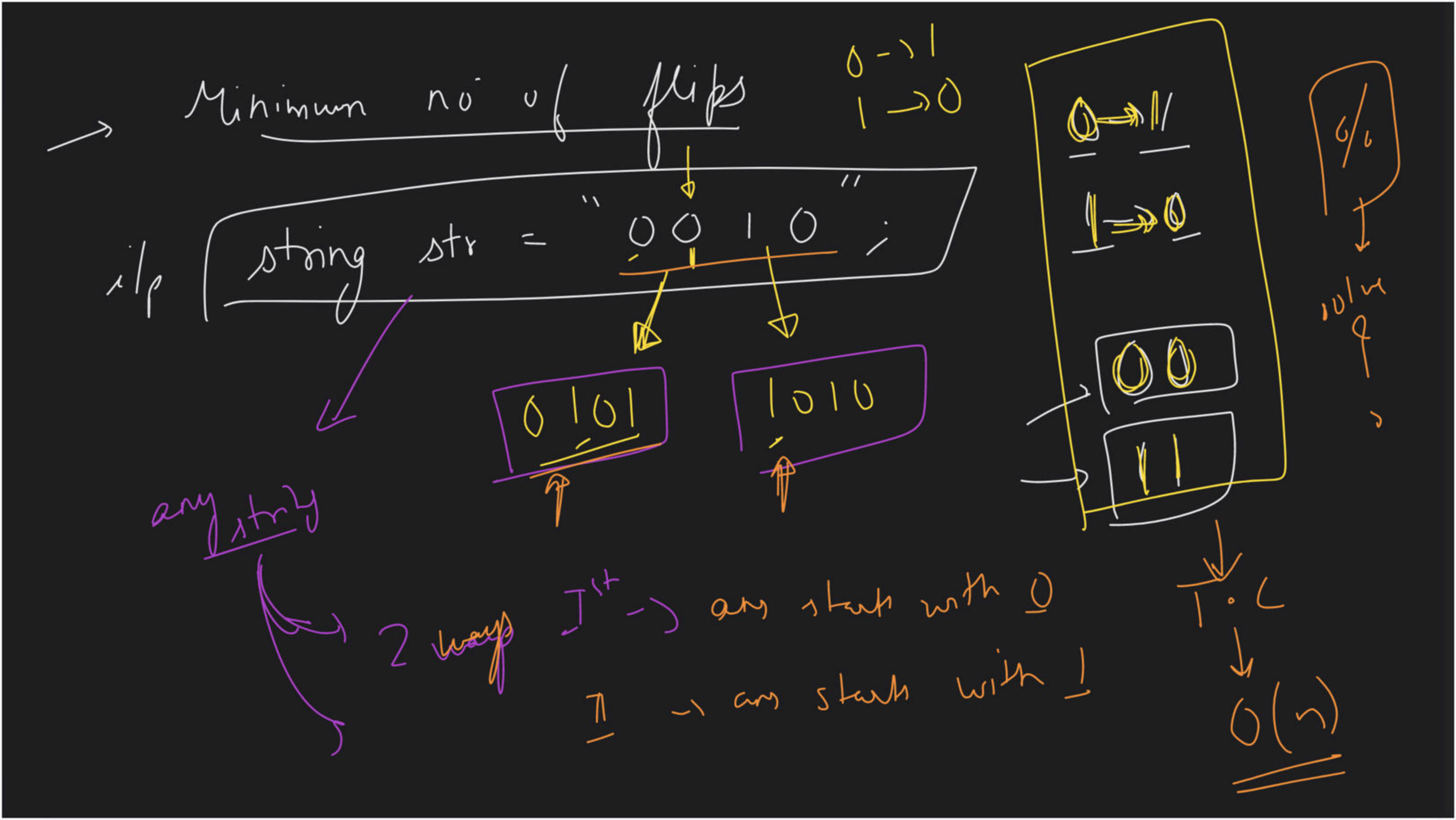


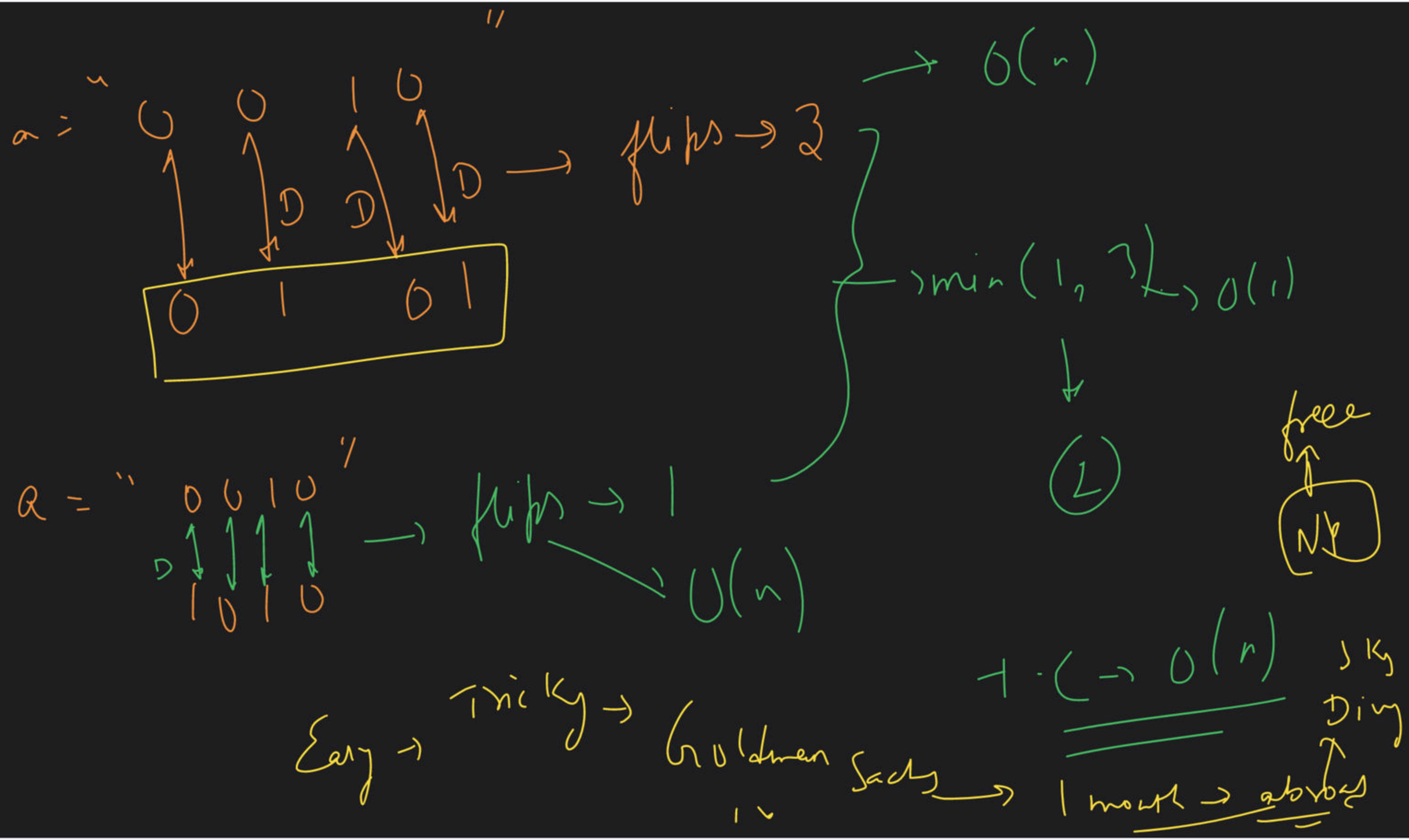
char ch = 'a'

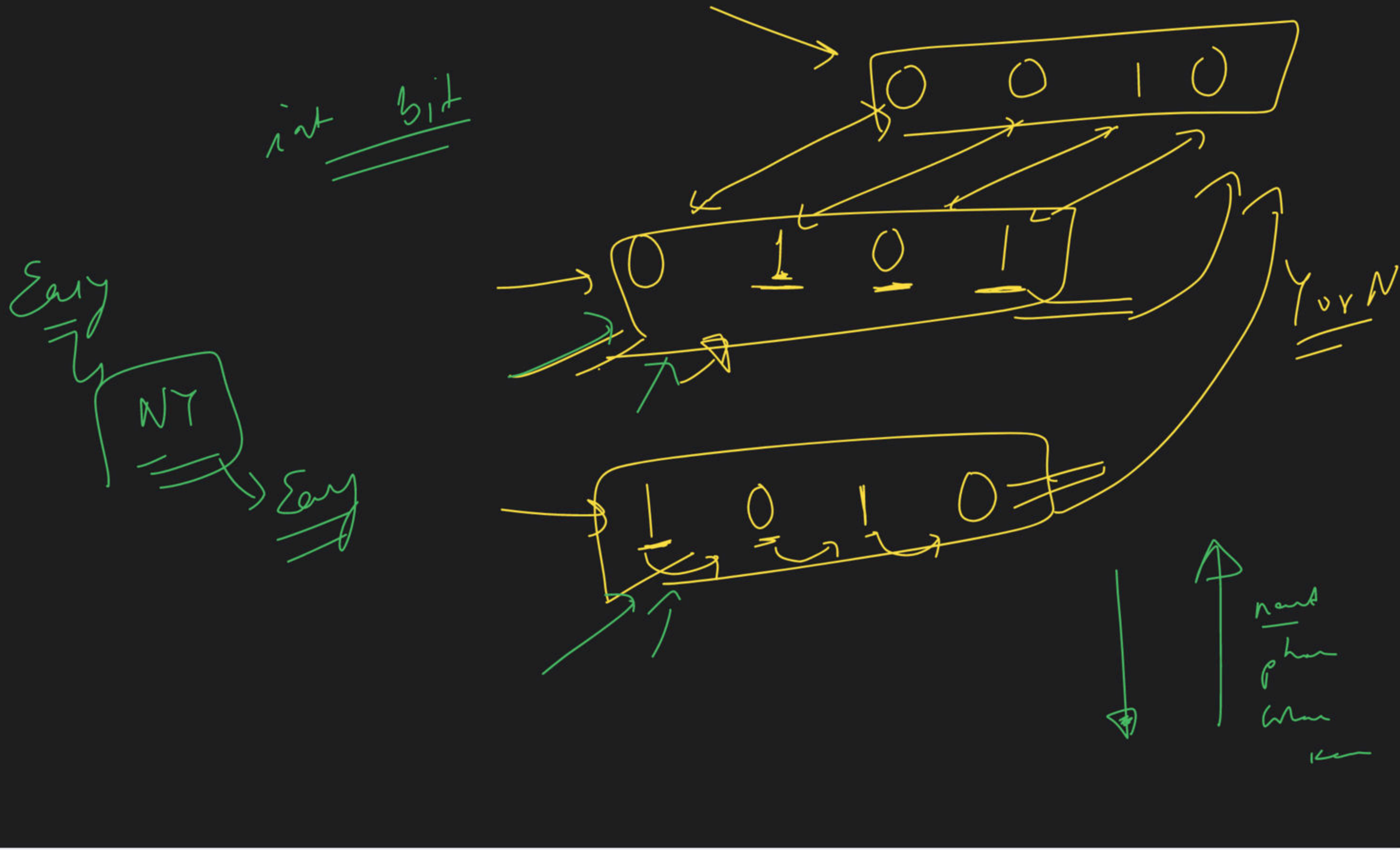
int index = ch - 'a'

smallest


map < char, int > m;
m[ch]++


Minimum no of flips
0 -> 1
1 -> 0
i/p
string str = "00101";
01101
10110
any string
2 ways
1st -> any start with 0
1 -> any start with 1
00
11
T.C
O(n)


a = "0 0 1 0"
flips → 2
→ O(n)
min(1, 2) → O(1)
Q = "0 0 1 0"
flips → 1
O(n)
free
Easy → Tricky → Guldmen Sachs
O(n)
Dijkstra


int bit
0 0 1 0
0 1 0 1
1 0 1 0
Easy
NY
Easy
Your
Normal

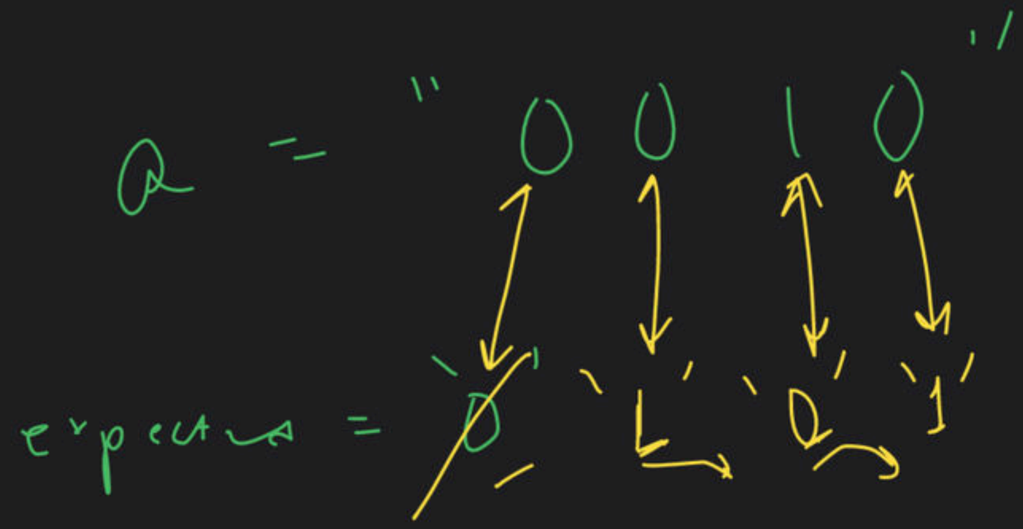
→ Code → [ ]

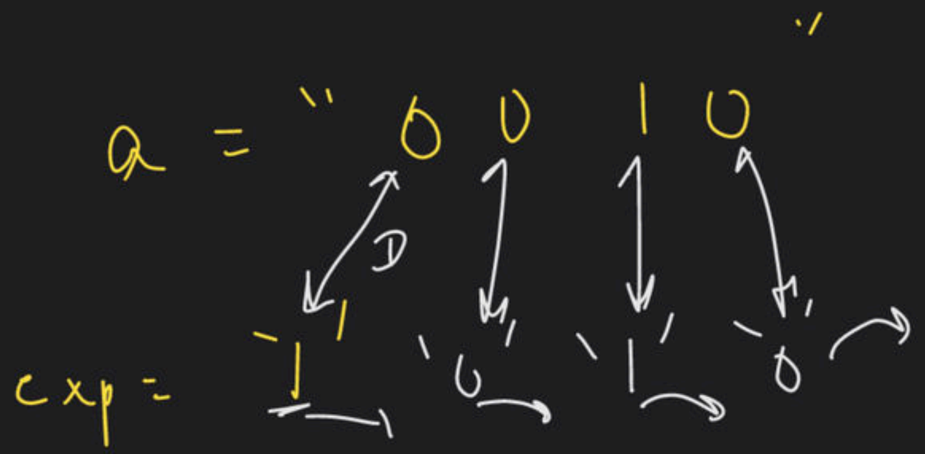
a = 0 0 1 0

a = "1011"

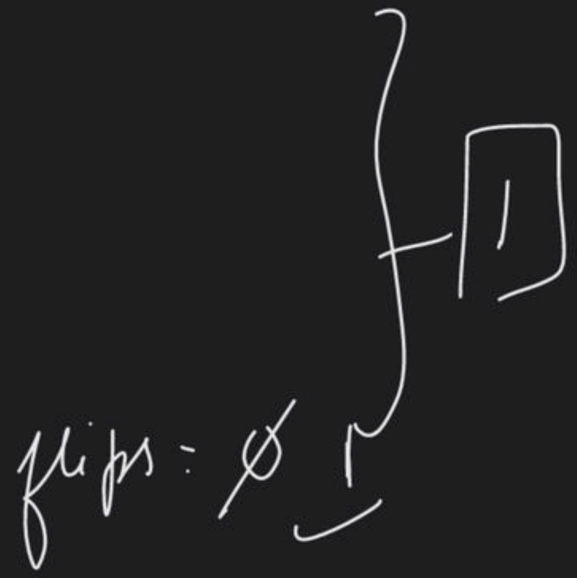
flips++

exp = 0

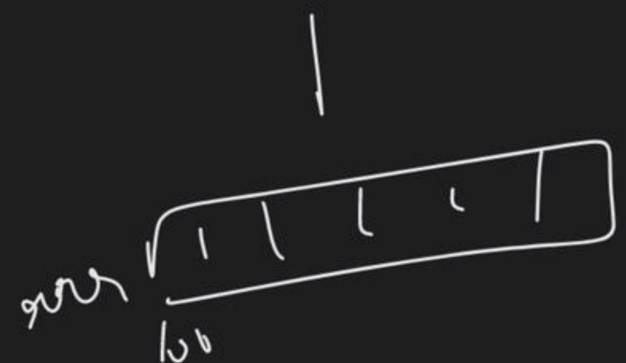




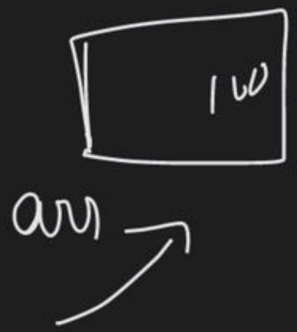
Flips = 0 1 2 3



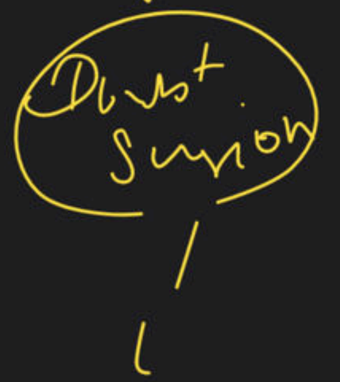
main()



reverse(str) char str[10]



follow-up



→ string a = "babbar"

→ string b = "abba"

→ check whether string B is present in string A

H/W

Easy

→ Check whether strings are rotation of each other or not

Amzn

string a = "bbarba" → 2 abbarb → babbar

string b = "babbar"

#1 string → Length → n (Brute) $O(n^2)$

1 → n → rotate (a)

↳ string → compare (b)

#2 string a = "bbarba"

① a + a = "bbarba" + "bbarba"

= "bbarbabbarba"

string b = "babbar" ② → cmp

→ Code

```
func ( string a , string b)
{
    if (a.length() != b.length())
        return false;

    string temp = a + a;
    if ( temp.find(b) == npos)
        return false;
    else
        return true;
}
```

string::npos

no position

no match

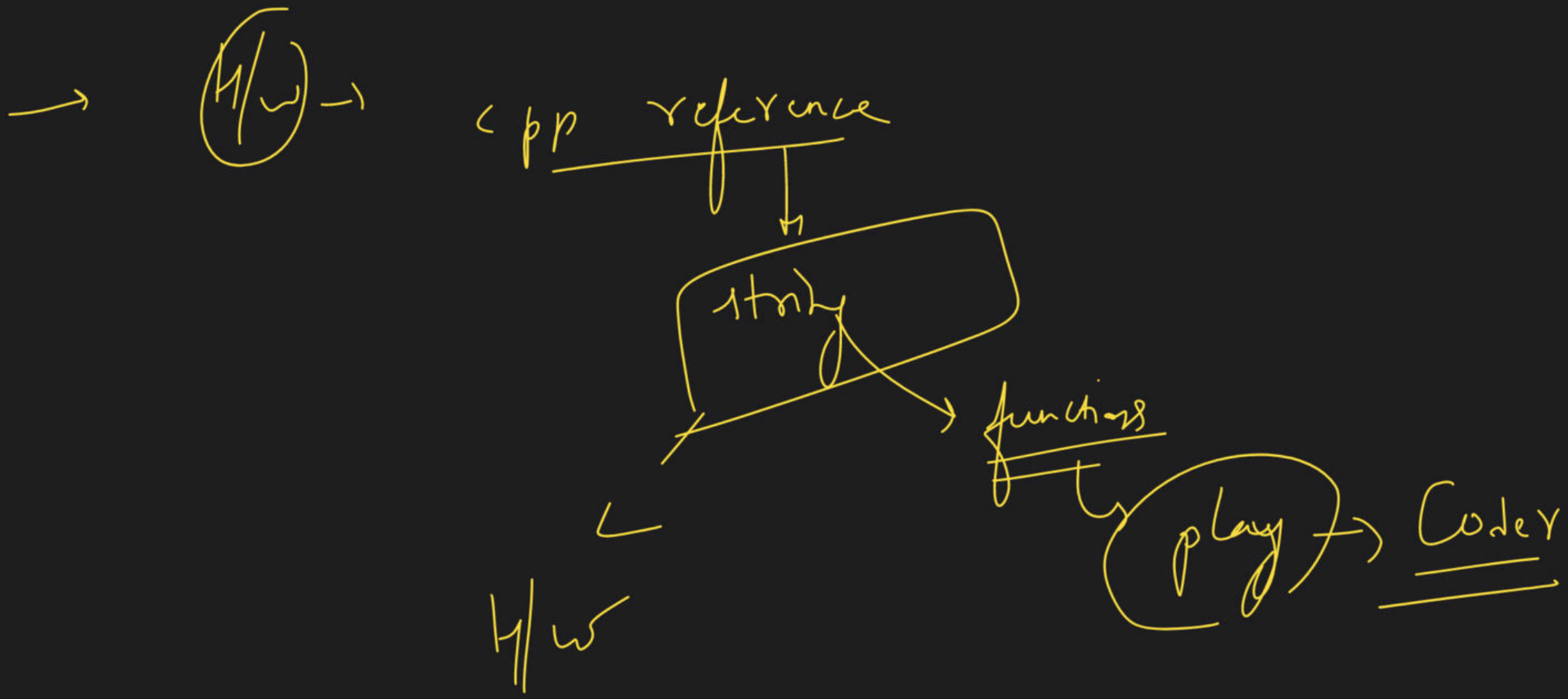
solution

reading

concat

temp.find(b) → no match

temp.find(b) → npos


H/w
cpp reference
string
functions
play
Coder
H/w

H/W → Assignment → solve

pakka promise

String Question sent on Discord → H/W → nahi aaye

5 video → 2.5 hr

news → X students → Surprise

1.5 hr

2.24 min

YT → solved

2 hr

CodeHelp → char Array & string