→ Recursion with Strings :-

H/W [4]

6th

Q-1 Reverse string

i/p → str = "babbar"

o/p → "rabbab" → using recursion

i/p -> "dipansh"

o/p -> "hsnapid"

i/p -> "Geetanjali"

"ilajnateeh"

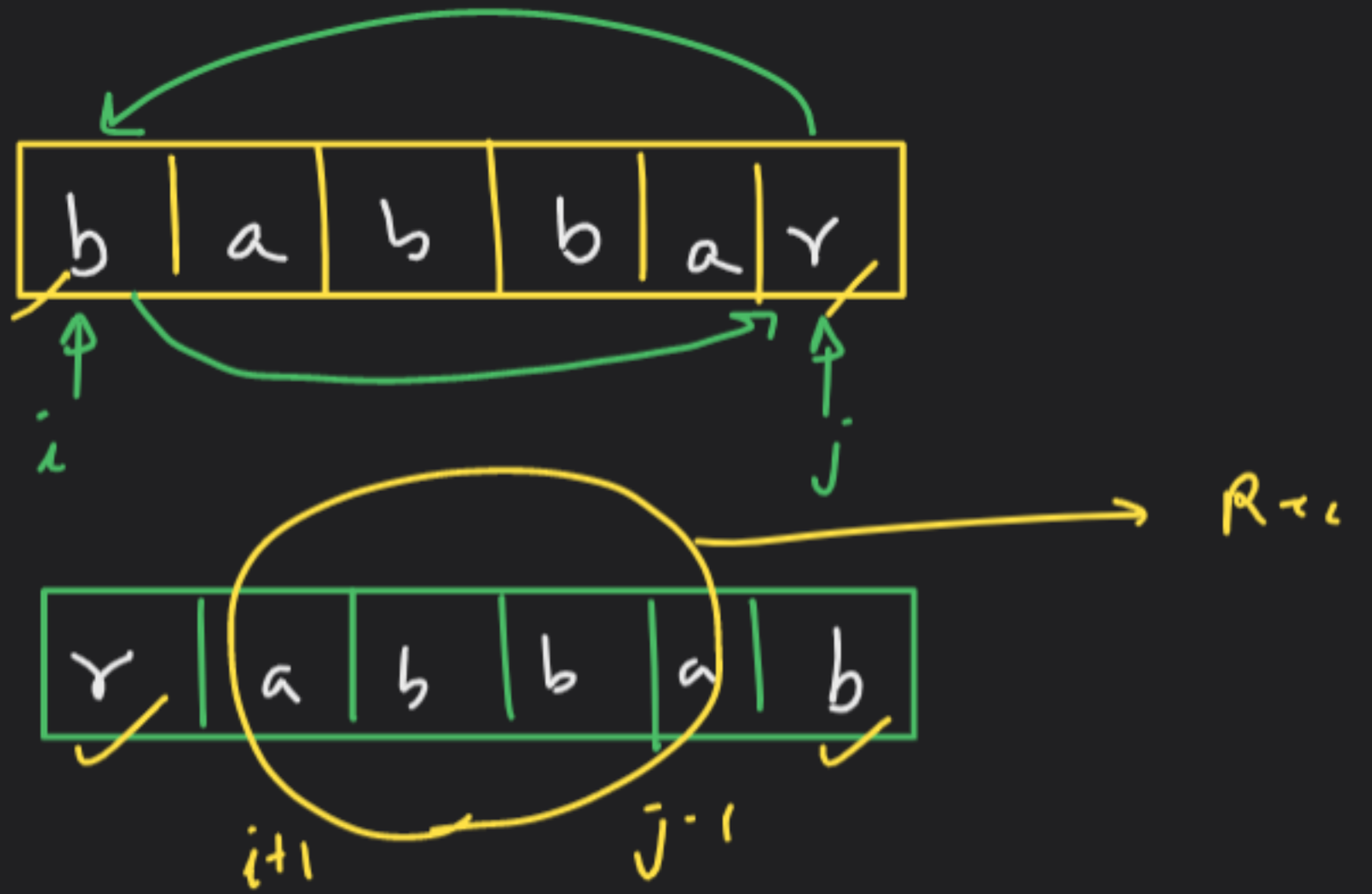
"Taniya"

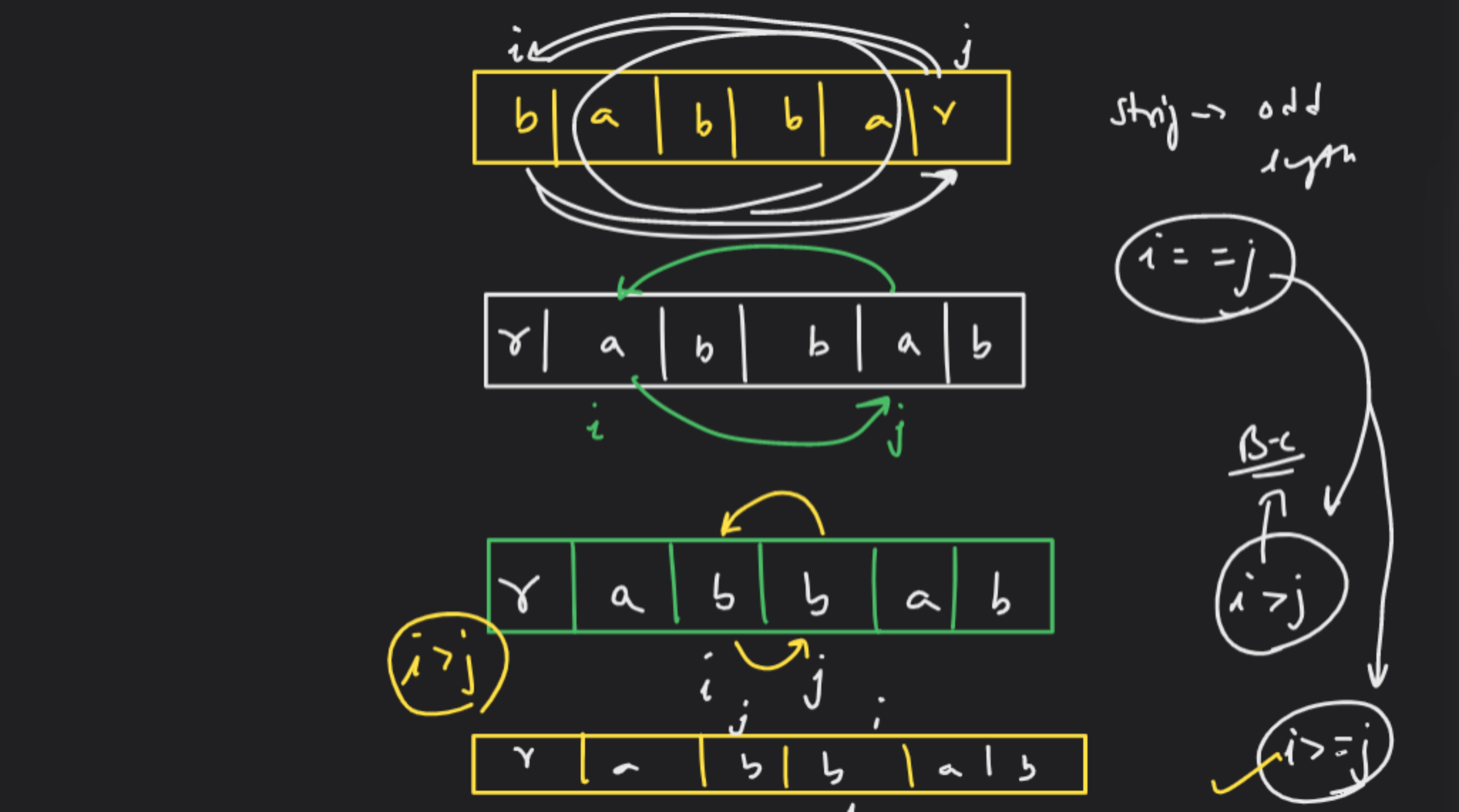
ayinaT

"Snehal"

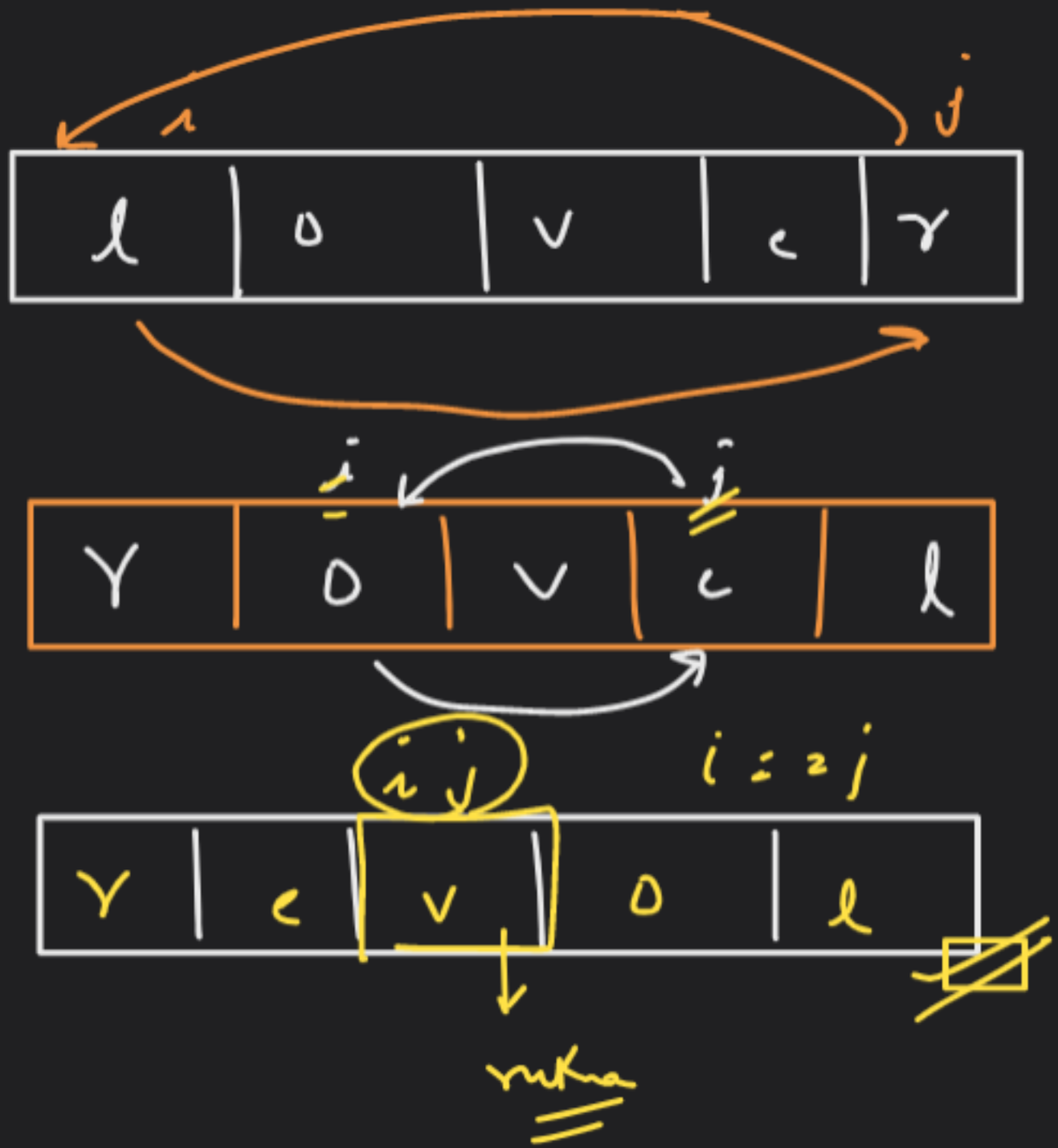
lahcnS

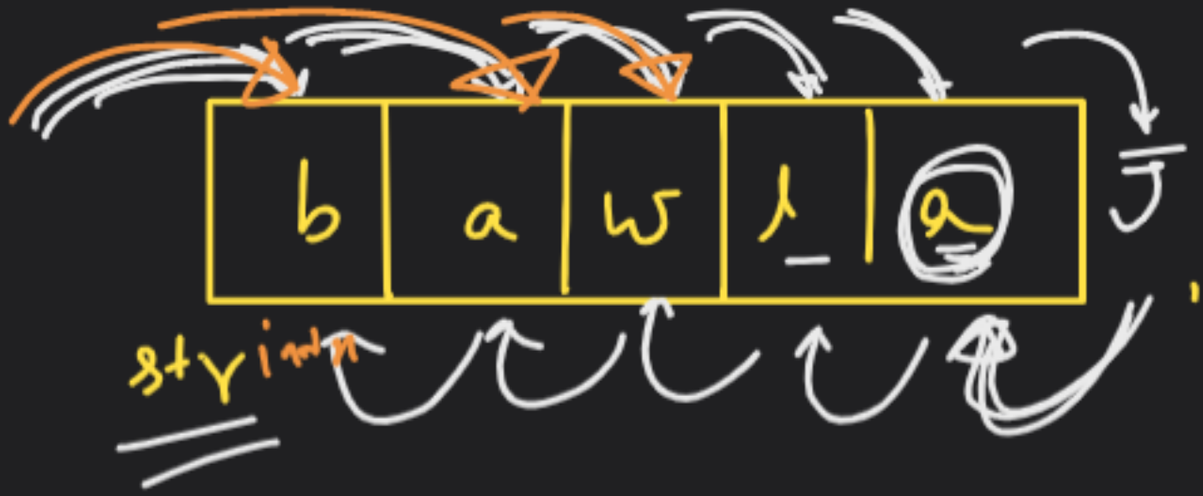
Approach :-




i
j
b a b b a r
str[i] -> odd length
r a b b a b
i
j
r a b b a b
i > j
i
j
j
i
r a b b a b
i == j
B-C
i > j
i >= j

```
void soln(string str, int i, int j)
{
    // Base case
    if (i >= j)
        return;

    swap(str[i], str[j]);

    soln(str, i+1, j-1);
}
```


i
j
l o v e r
i
j
r e v o l
i = j
r e v o l

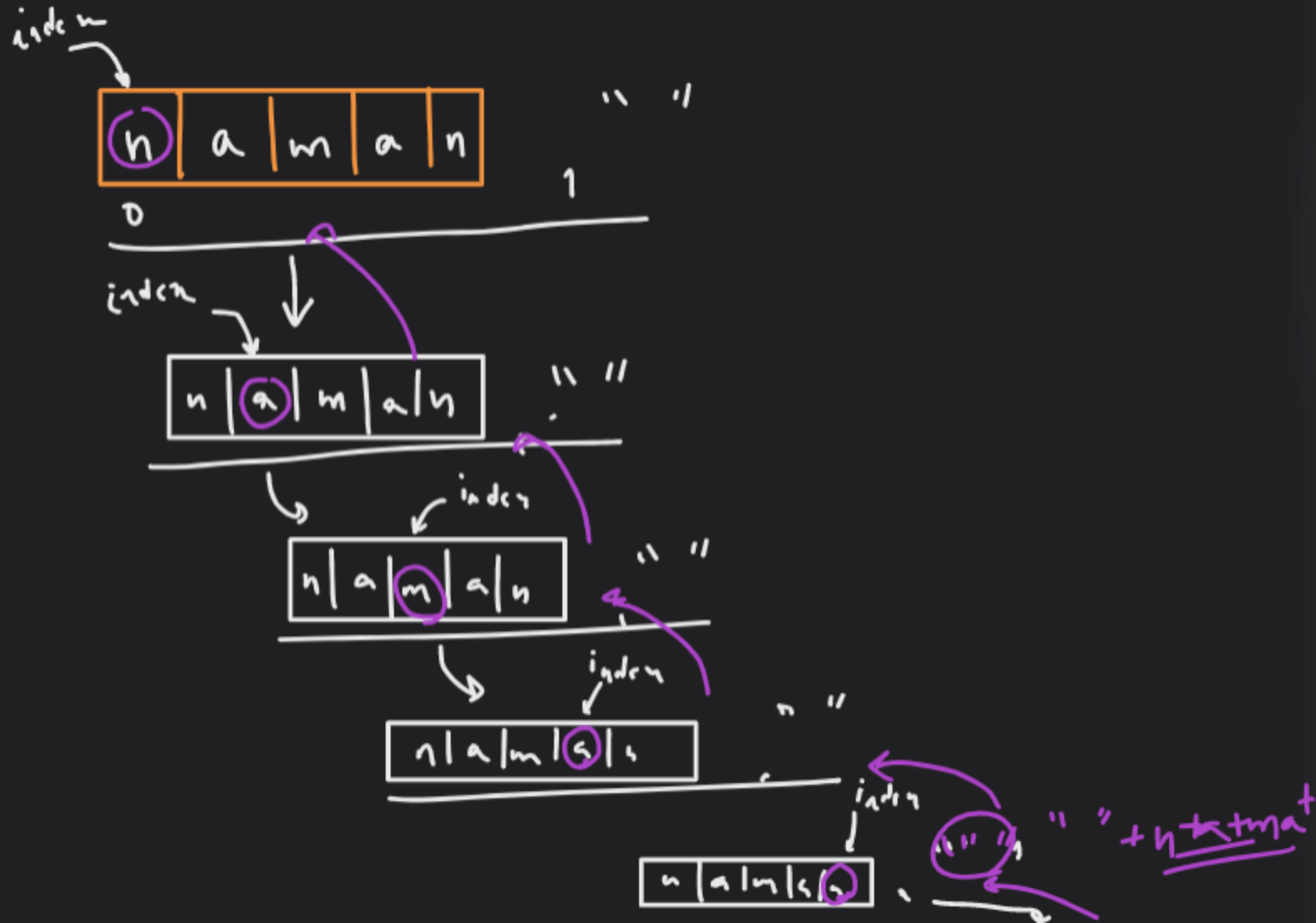


ans = "" + a + l + w + a + b

= alwab

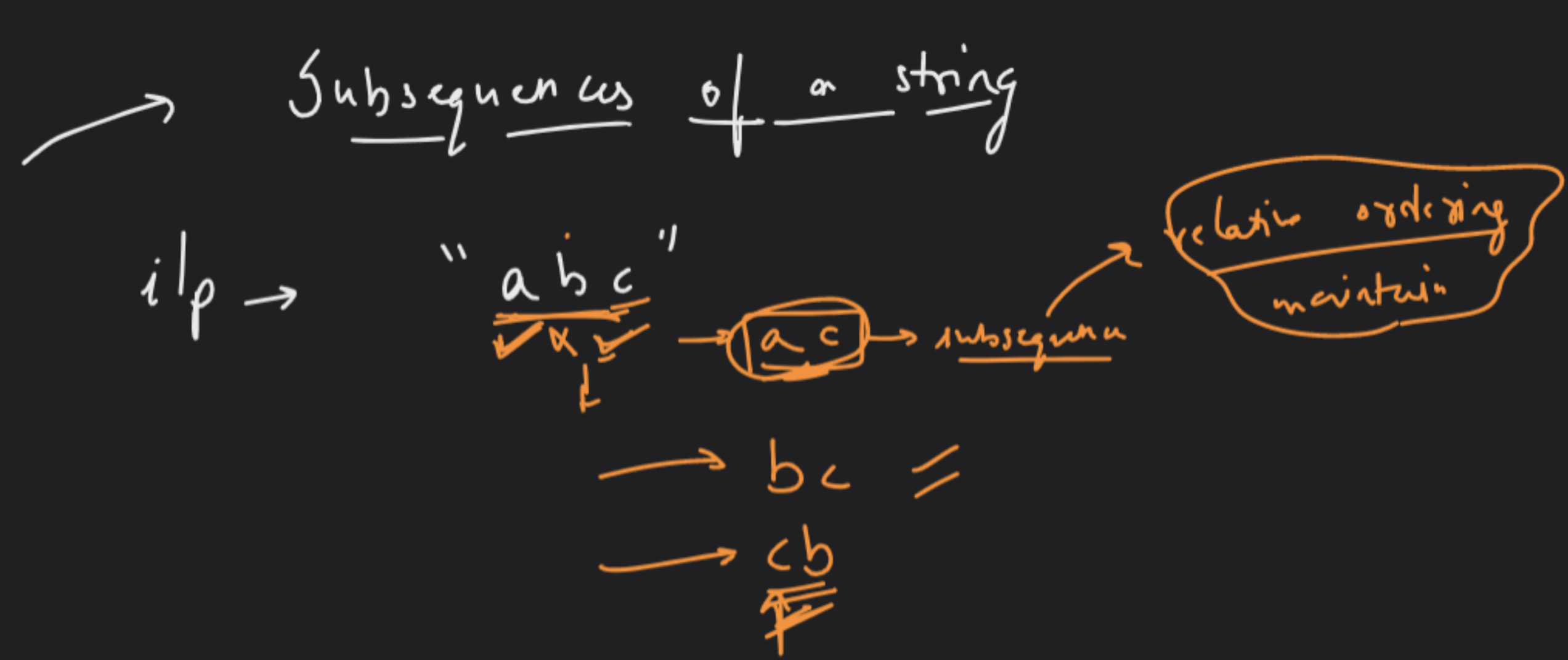
2 min

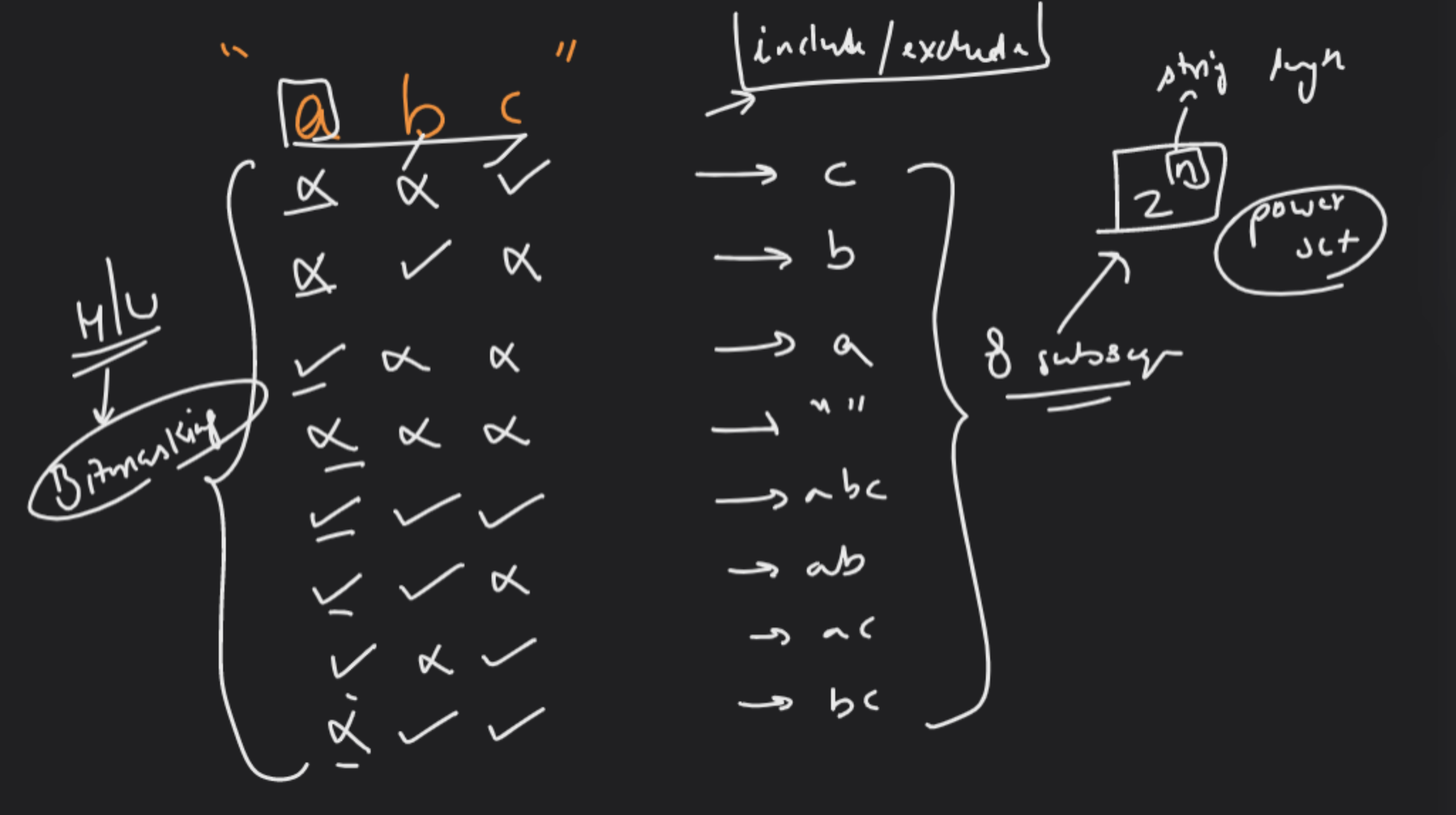
```
soln ( )
{   //ar
    if (index == size)
        return;

    soln( index+1);

    ans.push_back( str[index] );
}
```

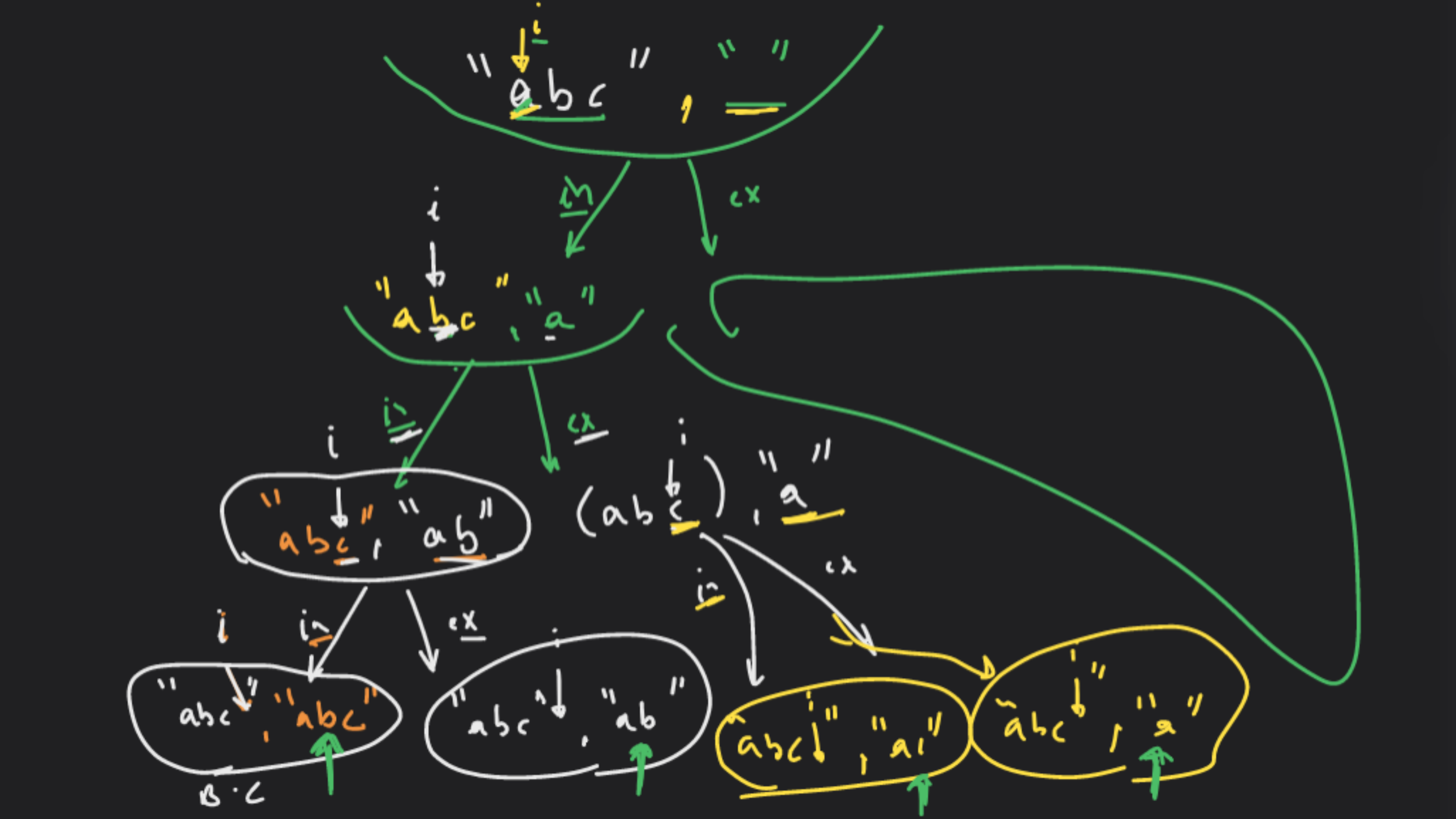

index
n a m a n
" "
0
1
index
n a m a n
" "
index
n a m a n
" "
index
n a m a n
" "
index
n a m a n
" "+n+tmp

exploration → String builder

Subsequences of a string




"
"
a b c
include / exclude
string length
n
2^n
power set
M/U
Bitmasking
subsets
→ c
→ b
→ a
→ ""
→ abc
→ ab
→ ac
→ bc

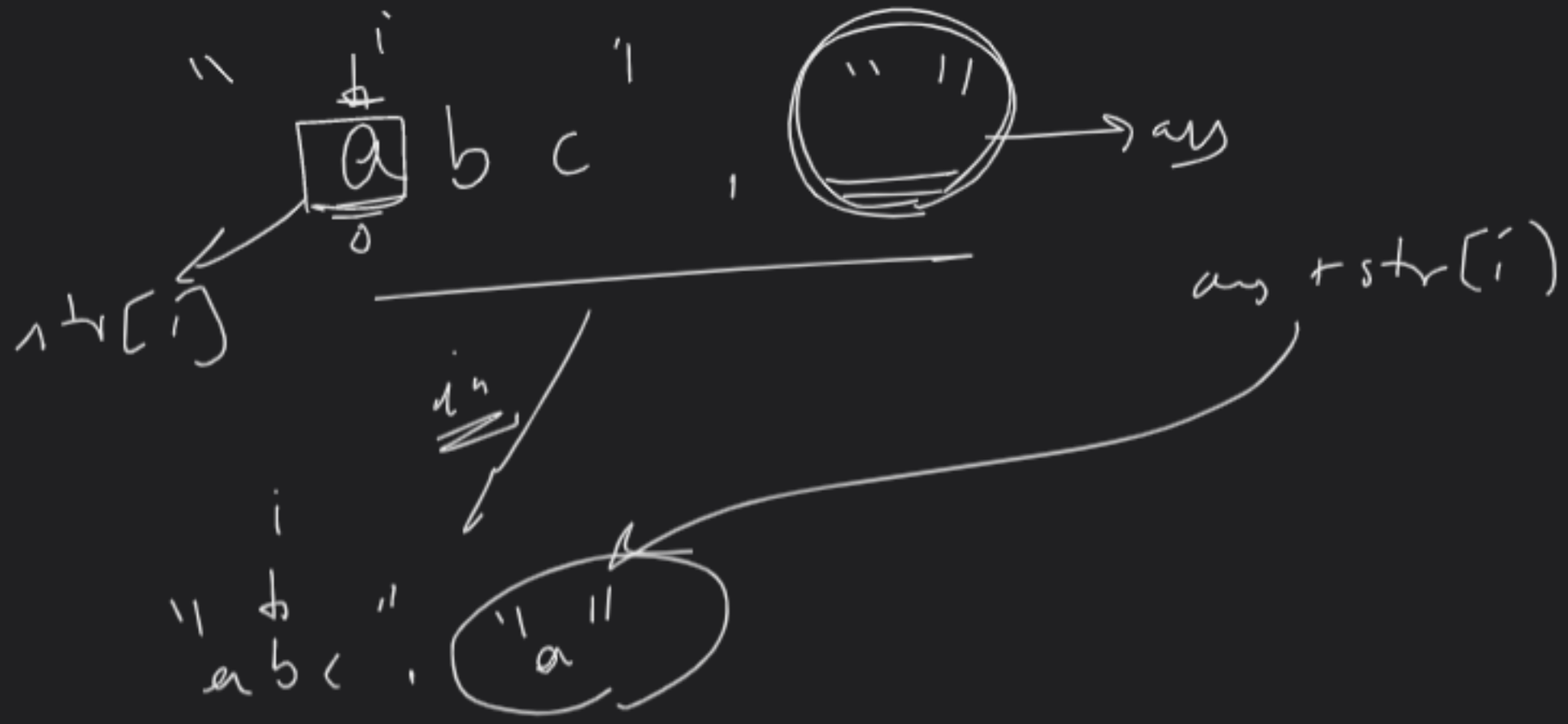

"abc" ""
i
in
ex
"abc" "a"
i
in
ex
"abc" "ab"
i
in
ex
(abc) "a"
i
in
ex
"abc" "abc"
"abc" "ab"
"abc" "a"
"abc" "a"
B·C

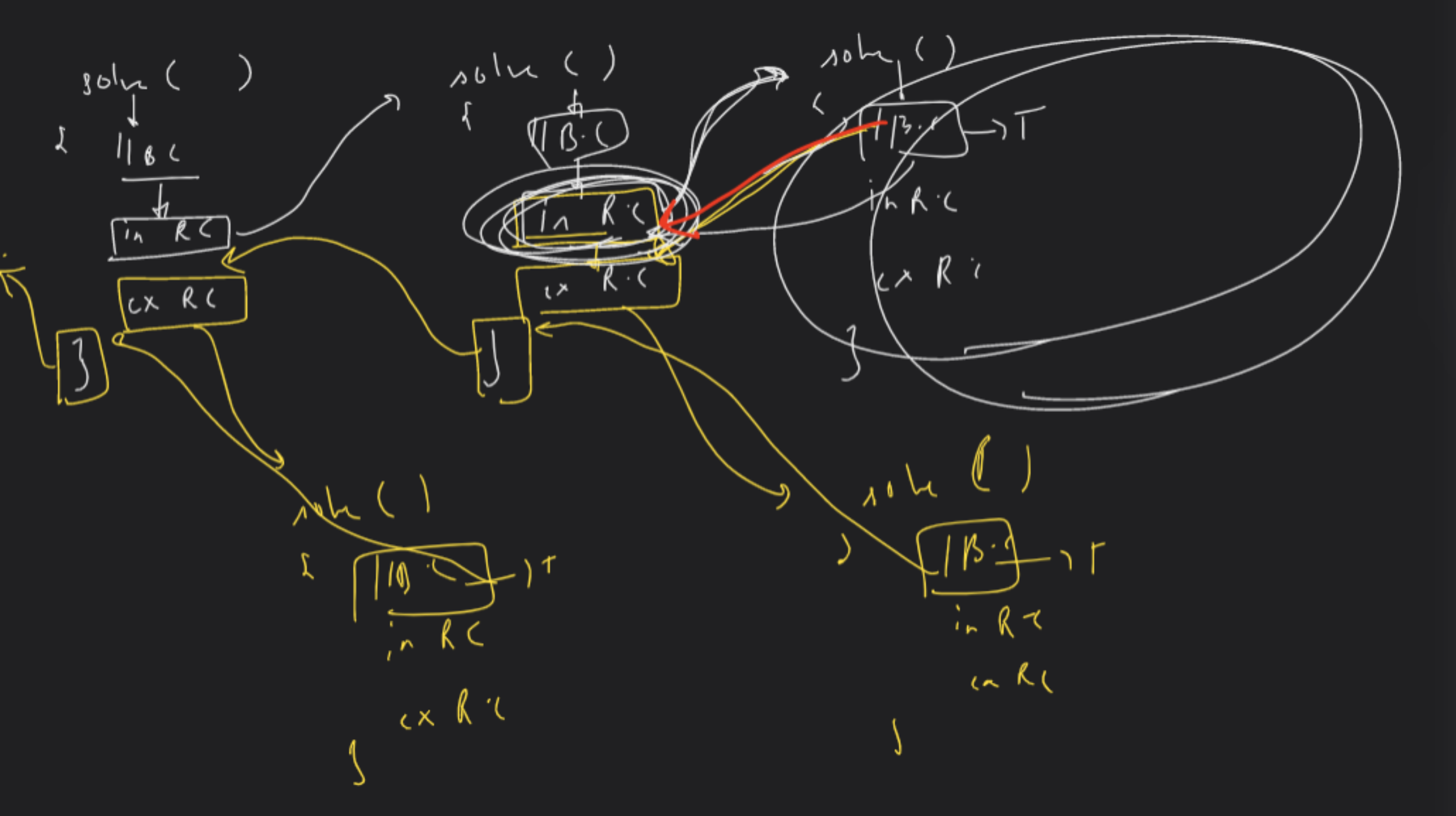
ans + str(indx)
ans
str(indx)
in
ex
"ab" , "a"
i==sz
ab → ab , a , b , ""

~~ans~~ ans = "abcd"

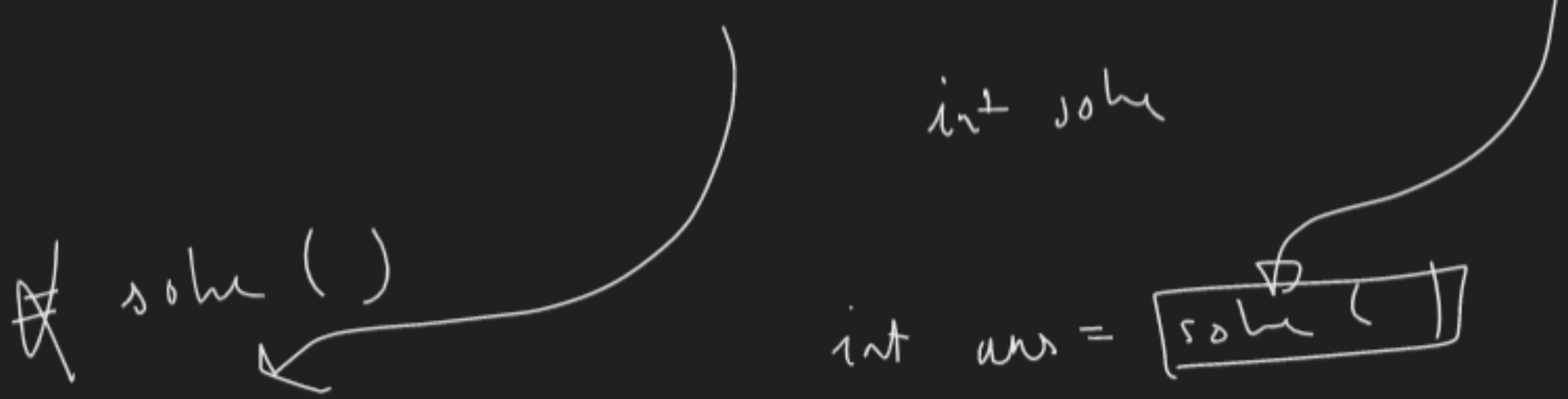
ans = ans + 'z'

ans.push_back('z')

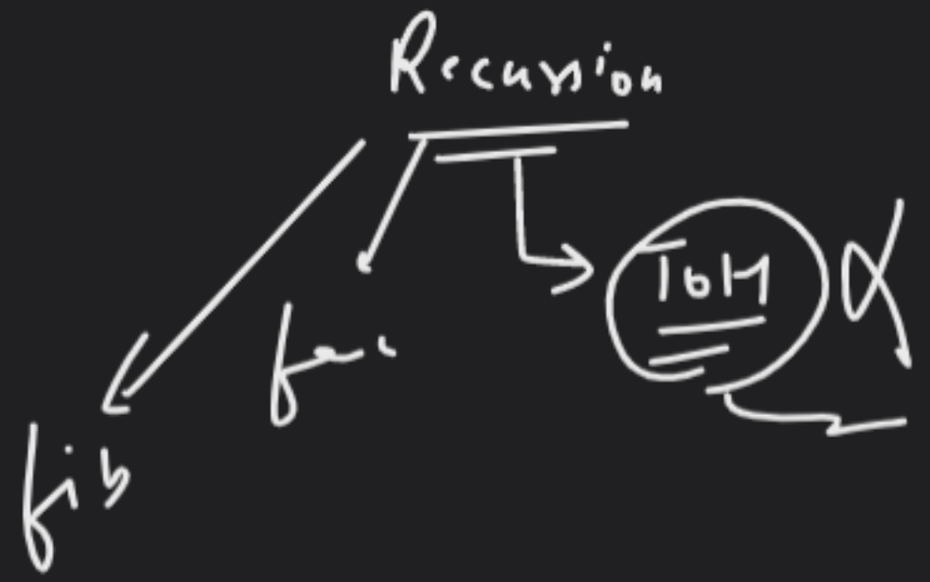

i
a b c
0
str[i]
ans
ans + str[i]
i
a b c
a


solve ( )
{
B.C
in RC
cx RC
}
solve ( )
{
B.C
in RC
cx RC
}
solve ( )
{
B.C
T
in RC
cx RC
}
solve ( )
{
B.C
T
in RC
cx RC
}
solve ( )
{
B.C
T
in RC
cx RC
}

void solve ( )



→ Q 3



→ add 2 strings

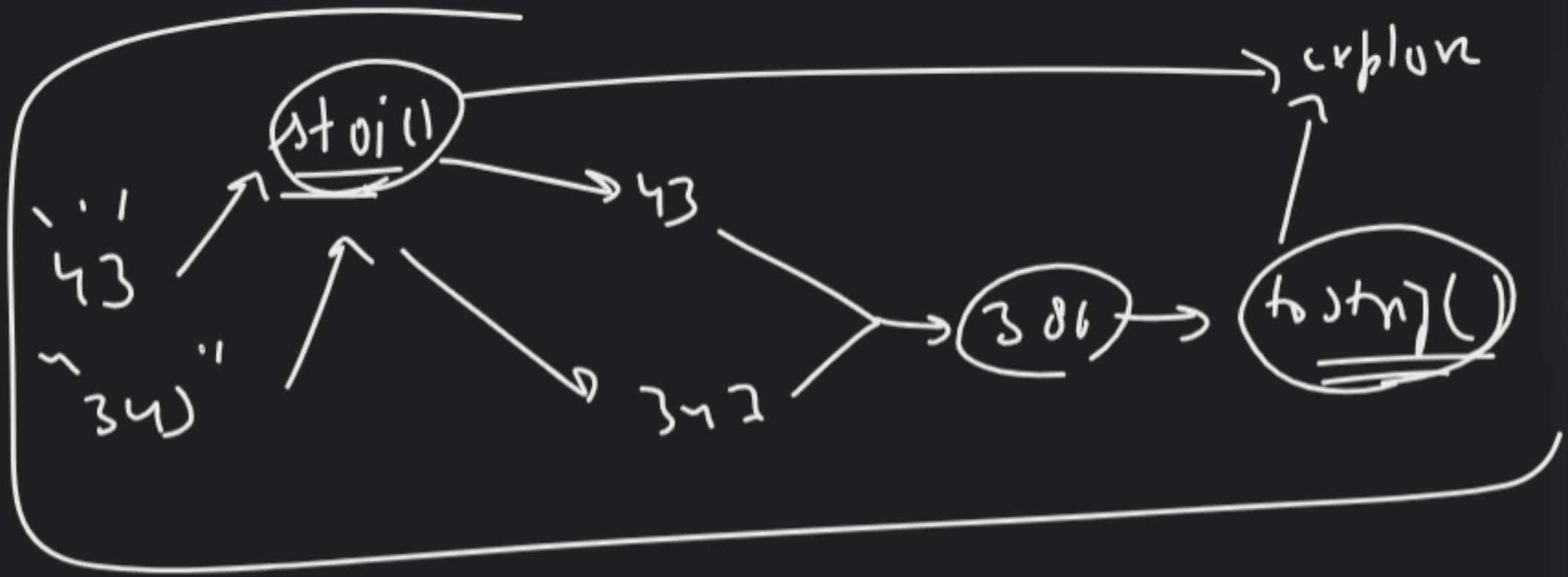
sum

i/p → a = "43"

b = "343"

```
  43
 343
-----
 386
```

o/p → a+b = "386"


stoi()
43
"34"
43
342
386
to_string()
cplus

str1 = "4 3" (i)

str2 = "3 4 3" (j)

carry = 0

str[i] + str[j] + carry

sum = 3 + 3 + 0

= 6

lastdigit = sum % 10

carry = sum / 10

rec (i-1 , j-1 , carry)

Base

i < 0

j < 0

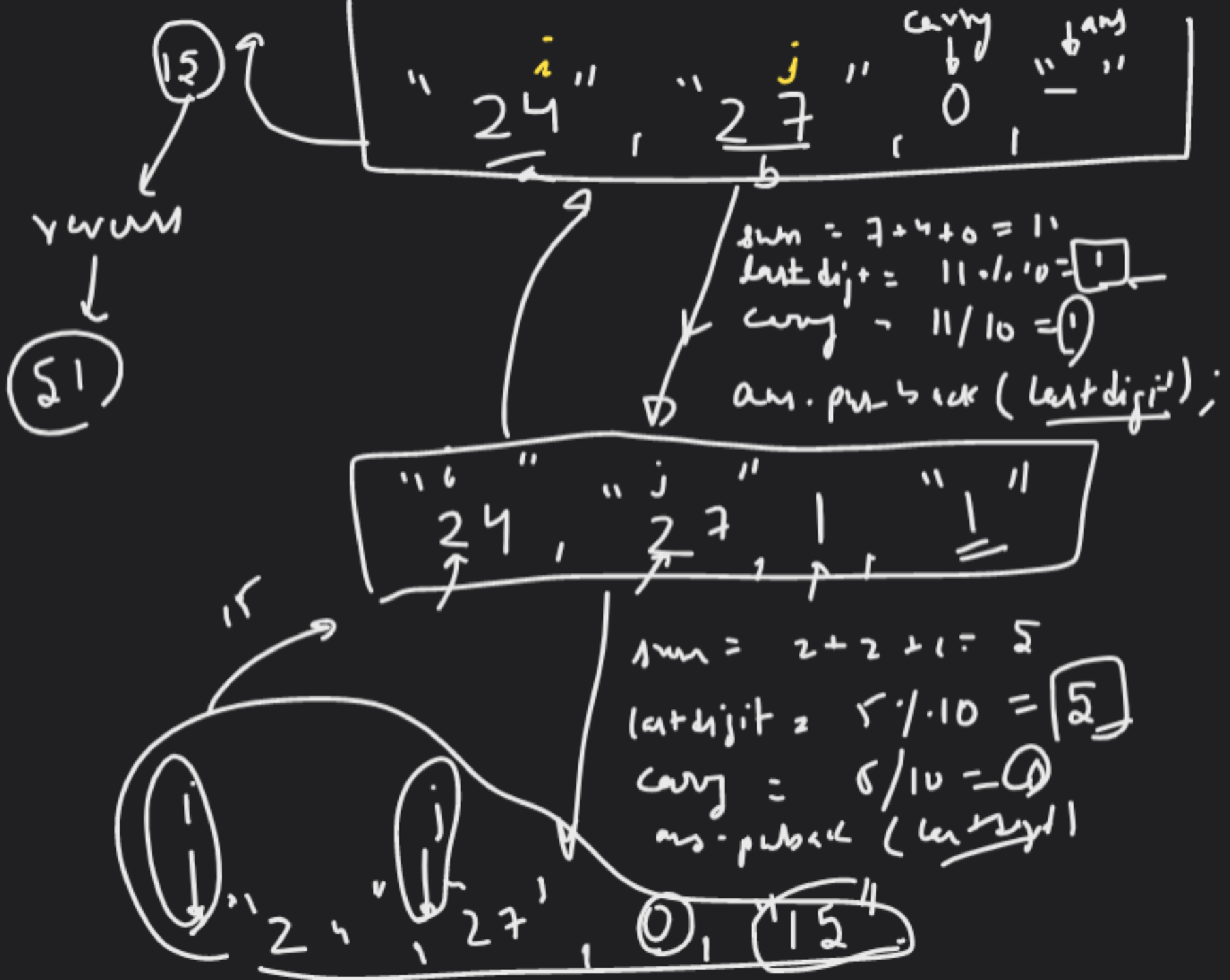
carry = 2 or 1

return

$a = 4$

$b = 243$

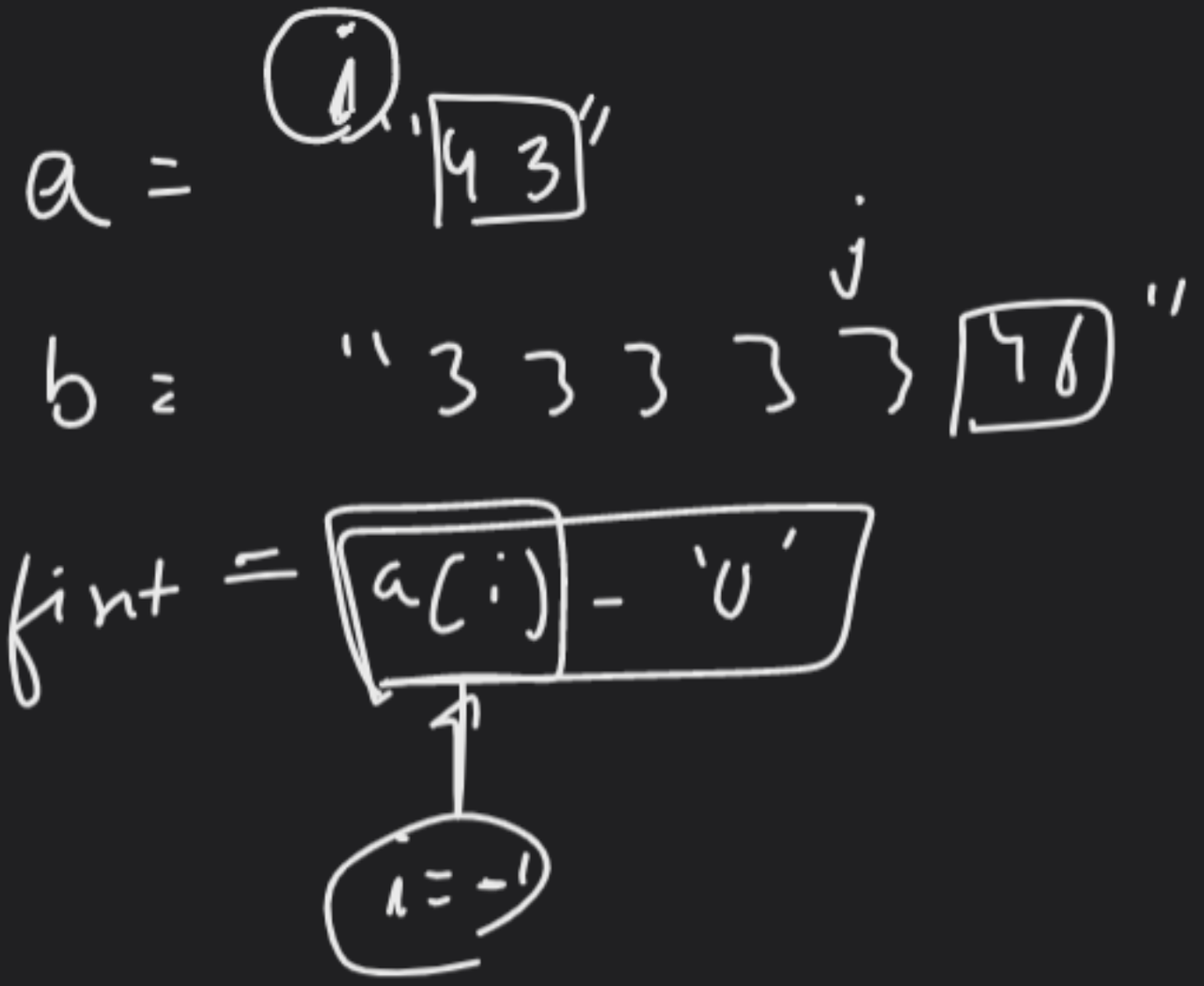
247

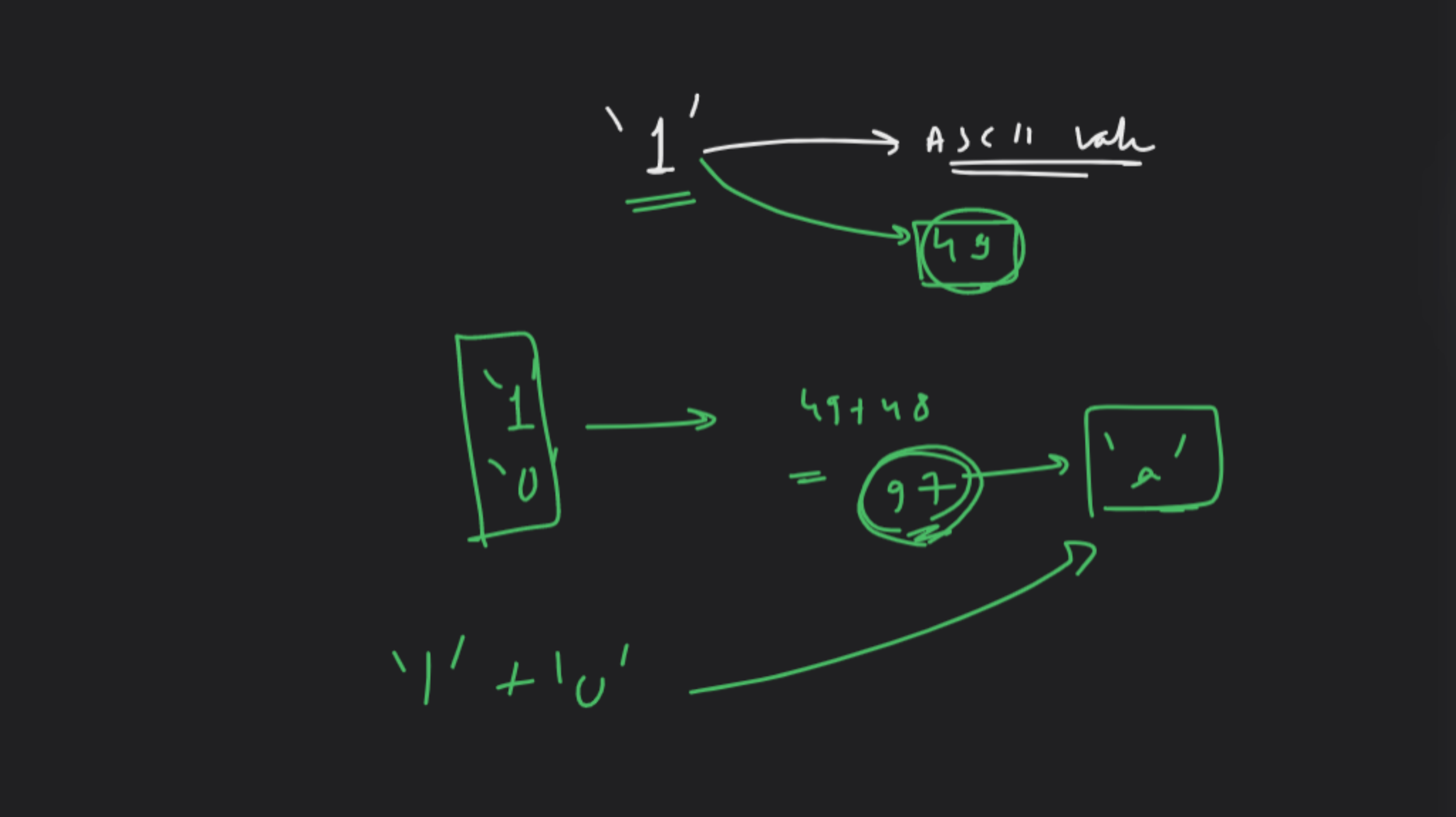

15
return
51
i
j
carry
ans
24
27
0
sum = 7+4+0 = 11
last digit = 11%10 = 1
carry = 11/10 = 1
ans.push_back(last digit);
i
j
ans
24
27
1
1
sum = 2+2+1 = 5
last digit = 5%10 = 5
carry = 5/10 = 0
ans.push_back(last digit)
i
j
24
27
0
15

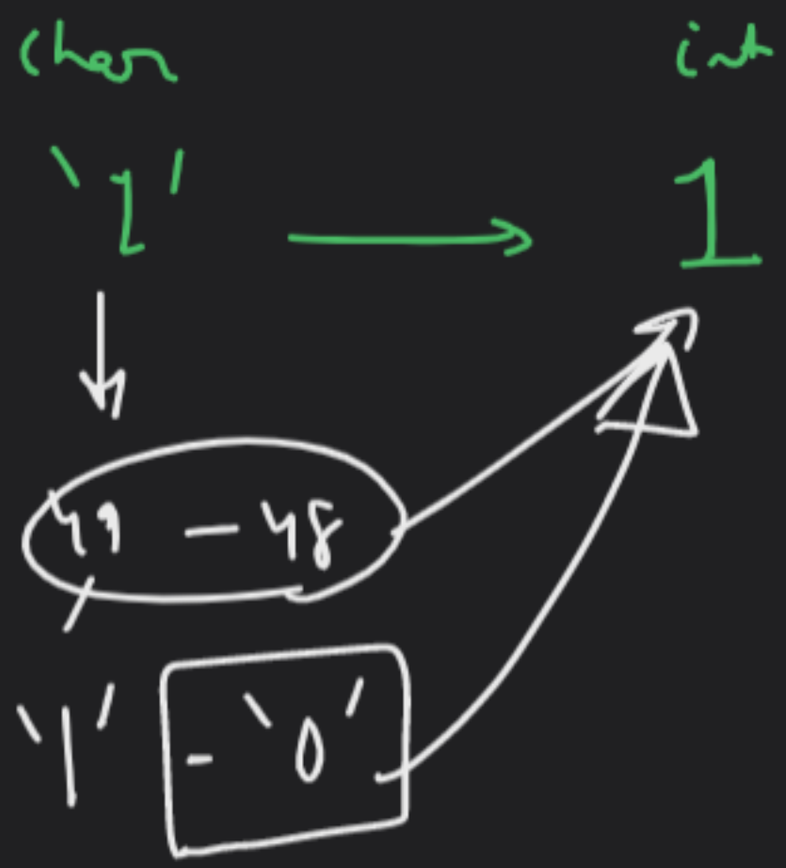
$a + (-b) \rightarrow a - b$

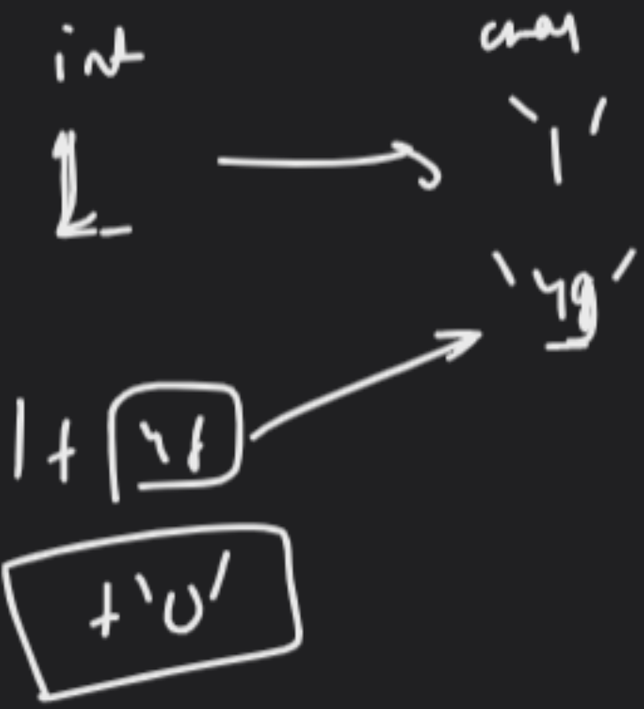
$(-a) + b \rightarrow b - a$

$(-a) + (-b) \rightarrow -(a+b)$


a =
i
"43"
b = "33337"
j
"76"
fint = a(i) - '0'
i = -1


'1'
ASCII value
49
'1'
'0'
49+48
=
97
'a'
'1' + '0'


char
int
'1'
1
'1'
- '0'


int
char
'1'
'9'
'1'
+ '0'