

"abc" → a, b, c, a  
bc, cb  
ac, ca, ...

include/exclude

# Subsequences & Doubt Clearing Session - LIVE

Special class

→ Rec / DP

→ Subsequences

→ Longest Inc. Subsequences

→ Longest Common Subseq

→ Longest Arithmetic Subsequence

→ Distinct Subsequences

Slides  
→ Monday

Rec  
→ 7.C  
→ 5.C

Thursday  
↑  
Extra class

Variant

Variant

→ Subsequences :-

relative ordering

"d d  
a b c d e"  
✓ x ✓ x x

"a c" ≠

ca  
f

len = 3

"	a	b	c	"
✓	x	x		
x	✓	x		
x	x	✓		
✓	✓	x		
x	✓	✓		
✓	x	✓		
✓	✓	✓		
x	x	x		

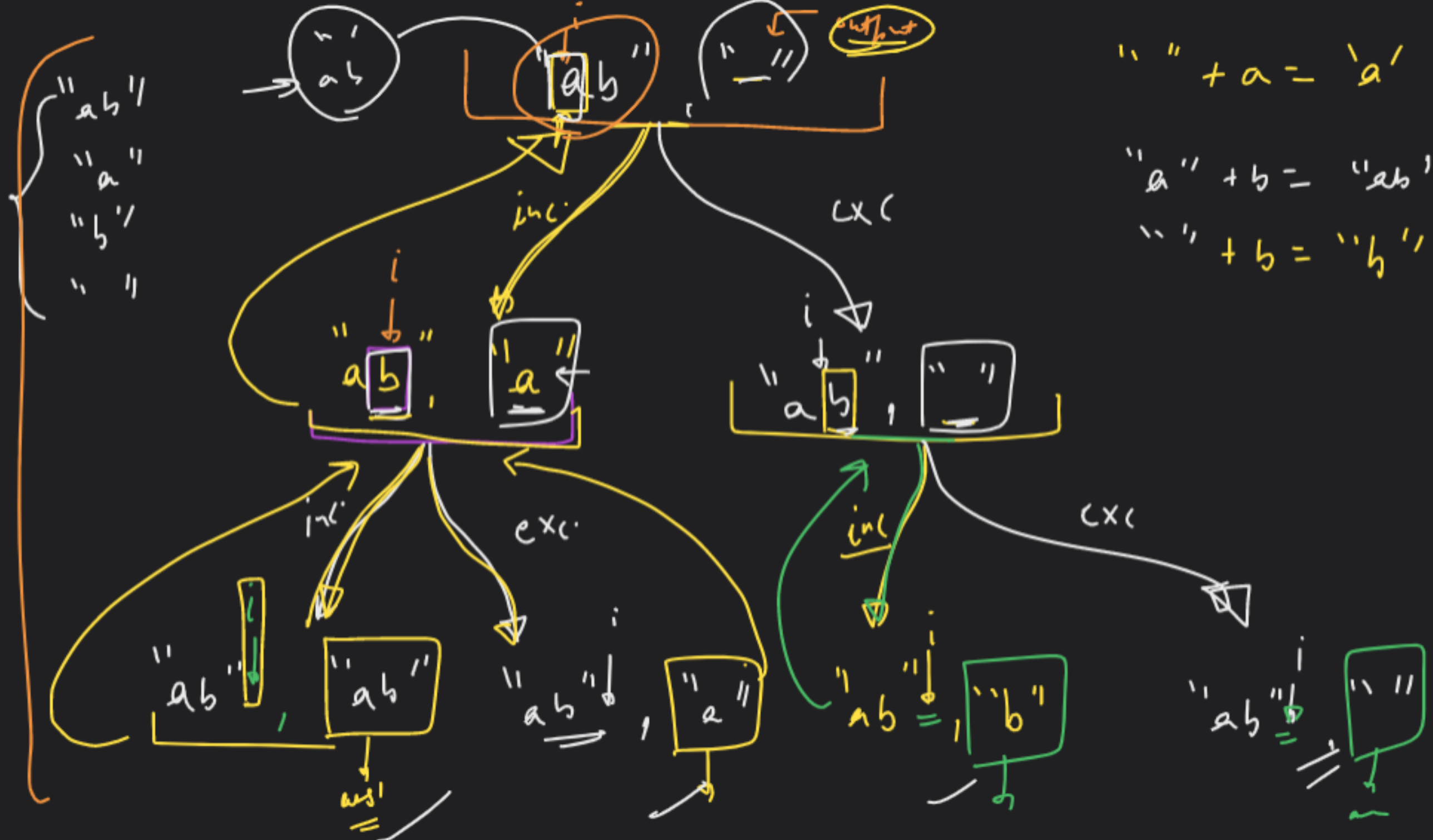
$$2^3 = \{$$

a  
b  
c  
ab  
bc  
ac  
abc  
" "

str  $\rightarrow$  length  $\rightarrow n$

$$2^n$$

subsequences



" " + a = 'a'  
"a" + b = "ab"  
"" + b = "b"

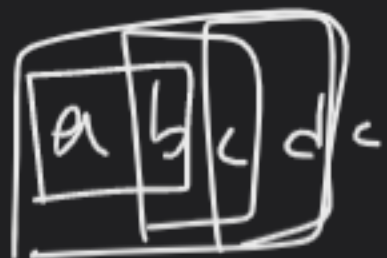
```

solve ( str , i , output )
{
    // B.C
    if ( i == str.length() )
    { // base case / print
      return;
    }

    // i.e
    solve ( str , i+1 , output + str[i] );

    // ex
    solve ( str , i+1 , output );
}

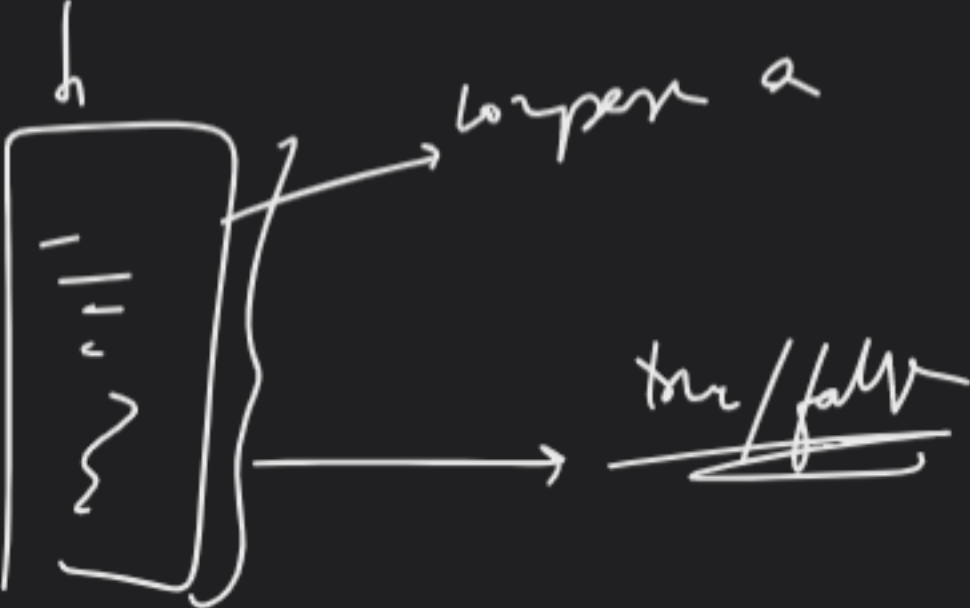
```



→ Tell me if a is a subsequence of b

a = "abc"      (true)  
          ↓    ↓    ↓  
b = "ahbgdc"

# → b → all subsequence



longest a

true/false



s =

"abc"

t =

"abbgdc"

solve()

no. of matched character

code

// inside

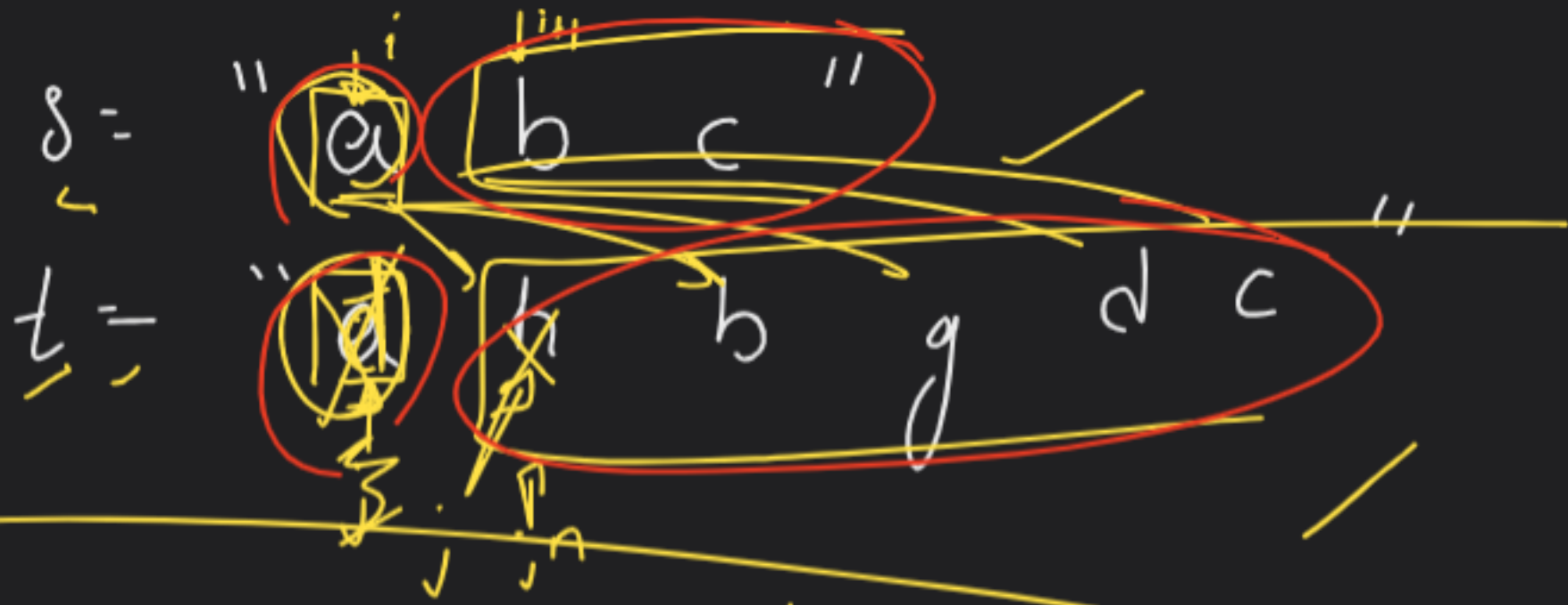
if (s[i] == t[j])

return

1 + solve(i+1, j+1, ...)

else

return solve(i, j+1, ...)



if ( $s[i] == t[j]$ )

return

1 +

solve( $i+1, j+1, \dots$ );

else

return

solve

( $i, j+1, \dots$ );

SF

③ "abc", "agbhd c"

2 min

① + [ ]

②

bc, gbhd c

bc, bhd c

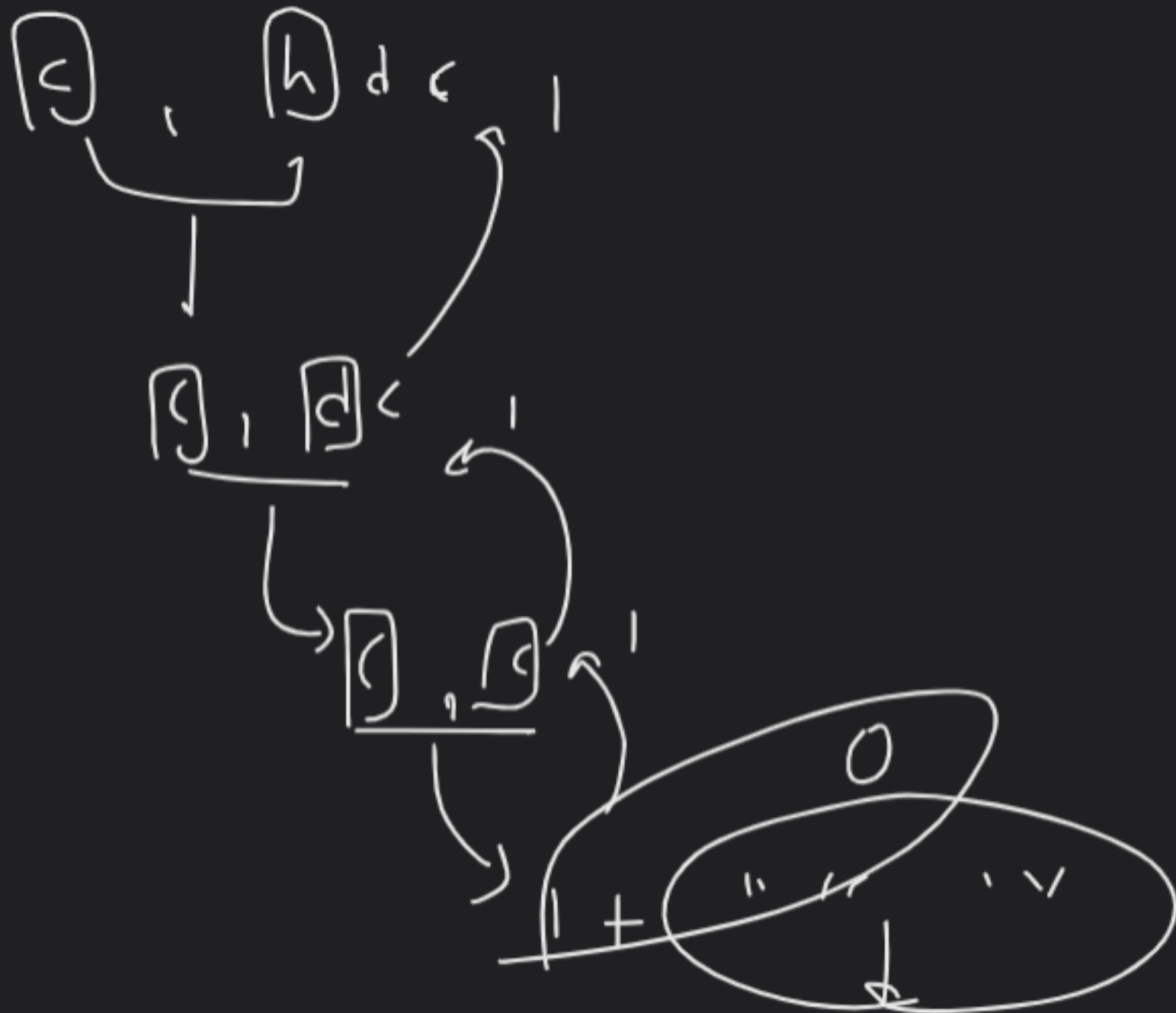
① + [c, h d c]

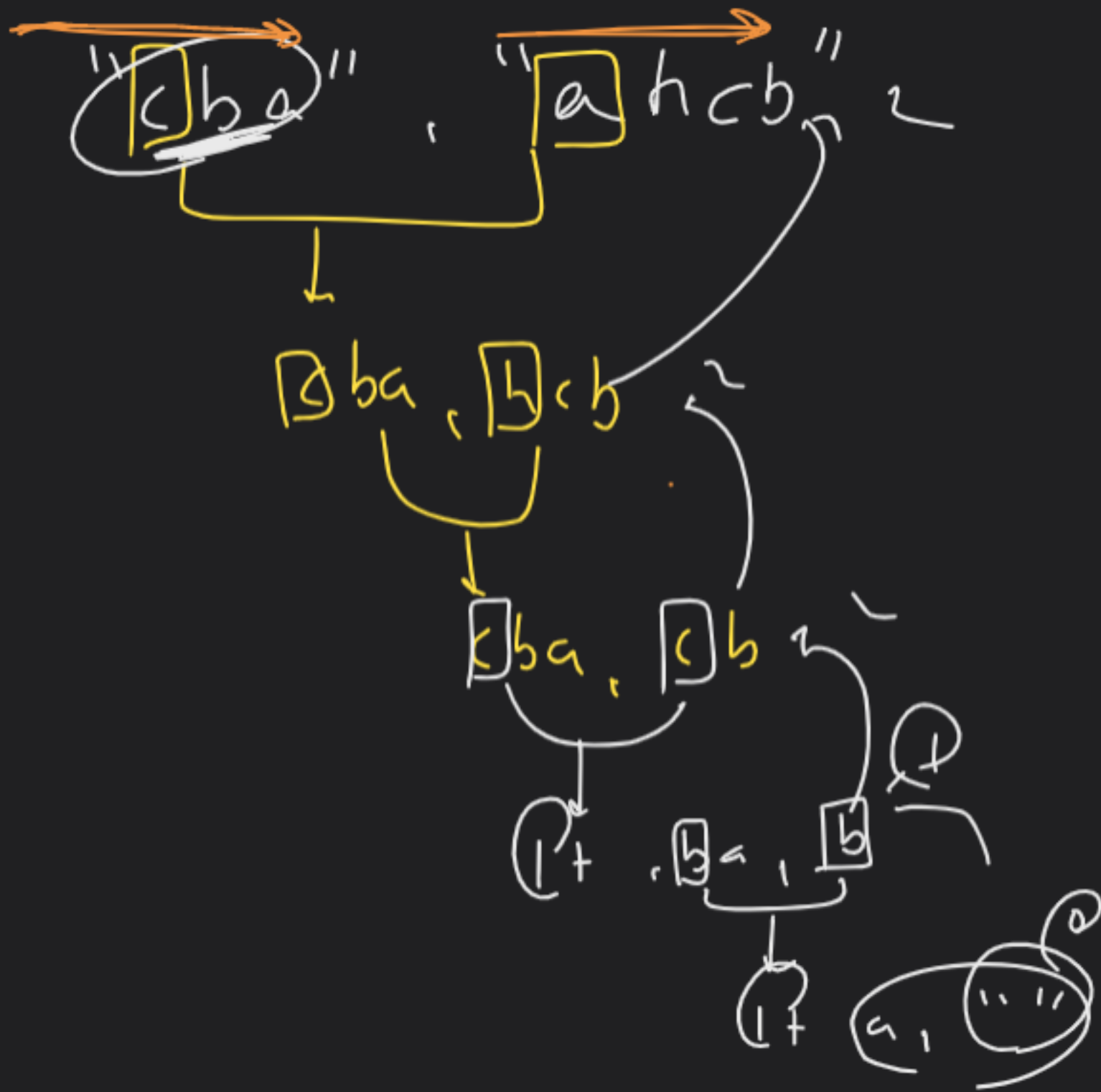
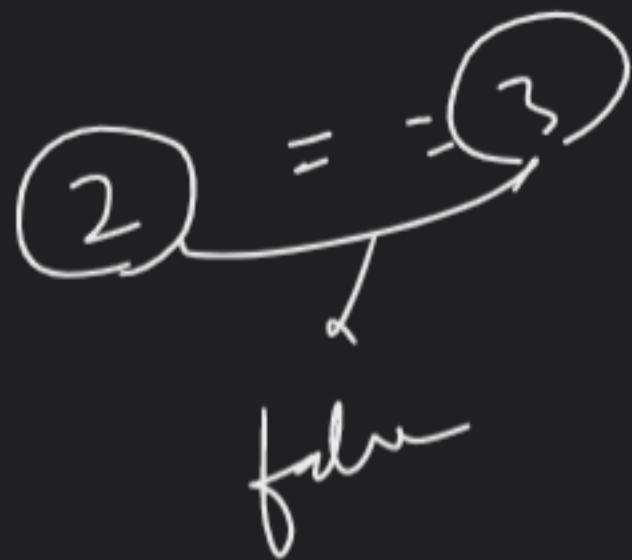
if (ans == s.length)

return true

else

return false





"  
abc  
"

'ahbgdc'  
⦿

Longest

Increasing

Subsequence

(2, 2)

trying  
inc

[0, 3, 1, 6, 2, 2, 7]

(2<sup>n</sup>)

[0, 3, 6, 7]

→

IS

[1, 2, 7]

[0, 1, 6, 7]

[3, 6, 7]

[0, 1, 2, 7]

[0, 2, 7]

[0, 6, 7]

$[0, 1, 0, 3, 2, 3], []$  ←  $em$

$1 > 0$

0 → 1 → 0

$[0, 1, 0, 3, 2, 3], [0]$

$[0, 1, 0, 3, 2, 3], [0, 1]$

$0 > 1$

$[0, 1, 0, 3, 2, 3], [0, 1, 0]$

$3 > 1$

$[0, 1, 0, 3, 2, 3], [0, 1, 0, 3]$

length of LIS



$[0, 1, 0, 3, \boxed{2}, 3], [0, 1, 3]$

$\alpha$   $i$   $cx$

$[0, 1, 0, 3, 2, \boxed{3}], [0, 1, 3]$

$\alpha$   $i$   $cx$

$[0, 1, 0, 3, 2, 3]$   $[0, 1, 3]$   
 $B \leq C$   $\text{len} 3$

include

ans ka last  
element, curr  
element se  
chutha ho

ab empty ho

prev = -1

include

prev = curr

prev = -1

or

ans[curr] > ans[prev]

curr  
[0, 1, 0, 3, 2, 3] prev = -1

curr  
[0, 1, 0, 3, 2, 3], prev = 0

curr  
[0, 1, 0, 3, 2, 3], prev = -1

in  
ex

curr  
[0, 1, 0, 3, 2, 3], prev = 1

incl

if (prev == -1 || arr[curr] > arr[prev])

1 + solve(curr + 1, prev);

// else

solve(curr + 1, prev)

take =  $\boxed{1}$



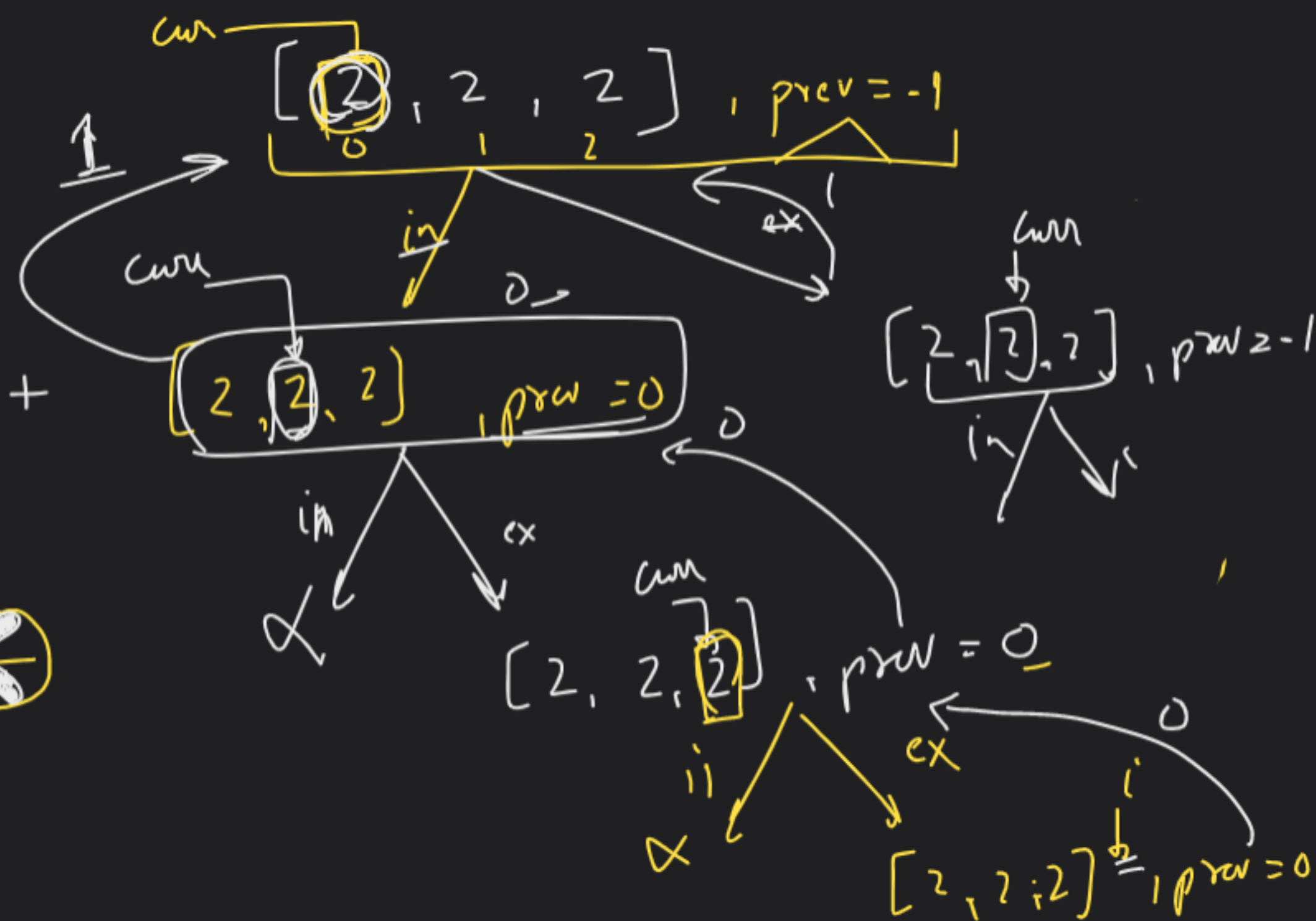
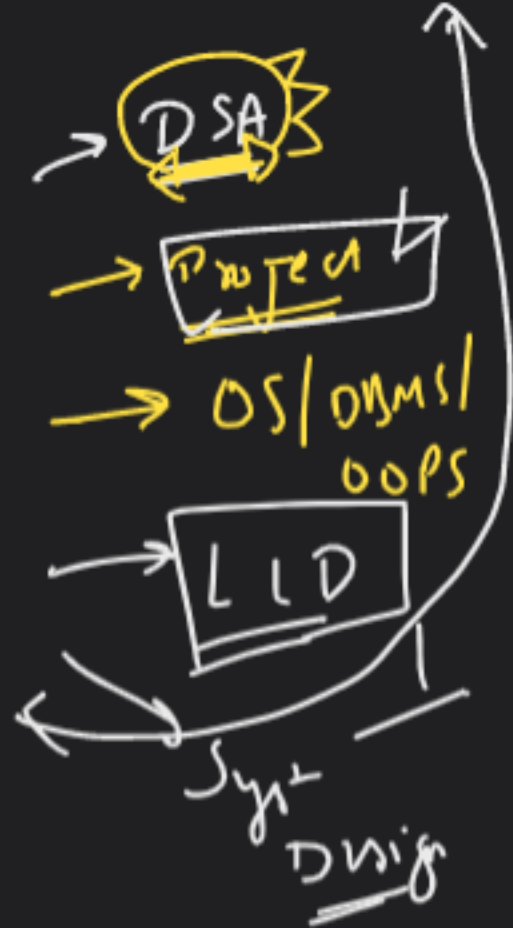
solve (cur, n, cur+1, cur)

Rec

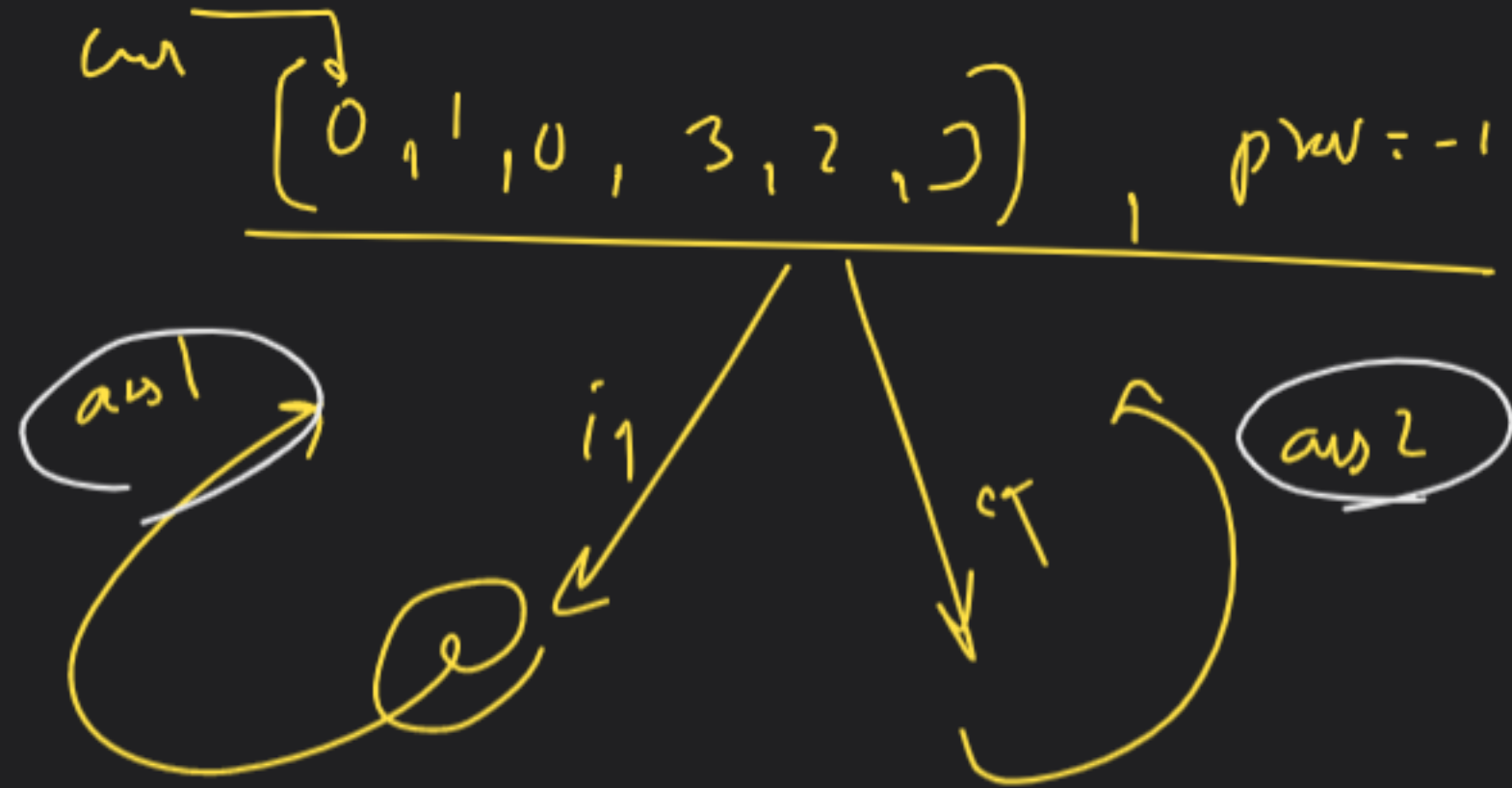


height of tree

Rec tree



On Campus ~~X~~



Longest Increasing Subsequence

$> \rightarrow 110K$

B.S

variants

4/5

LIS,  $\rightarrow$  with difference  $> 0$

adjacent element  $\neq 1$

(1 - 2 - 3 - 4)

# Longest Common Subsequence

i/p  $\rightarrow$  str1 = abc  
str2 = xyc

abc  $\rightarrow$

a

b

c

ab

bc

ac

abc

xyc  $\rightarrow$

x

y

c

xc

yc

xy

xyc

1  $\rightarrow$  1



>>>>

x x x

abc

abc

abce

abcn

abfd

abfd

acdef

acdef

5

str1 =  $\boxed{a}bcde$

str2 =  $\boxed{a}ke$



if (str1[i] == str2[j]) // match

ans = 1 + solve(i+1, j+1)

else

ans = max( $\frac{\text{solve}(i, j+1)}{\text{solve}(i+1, j)}$ )



$$1 + \text{soln}(i+1, j+1)$$



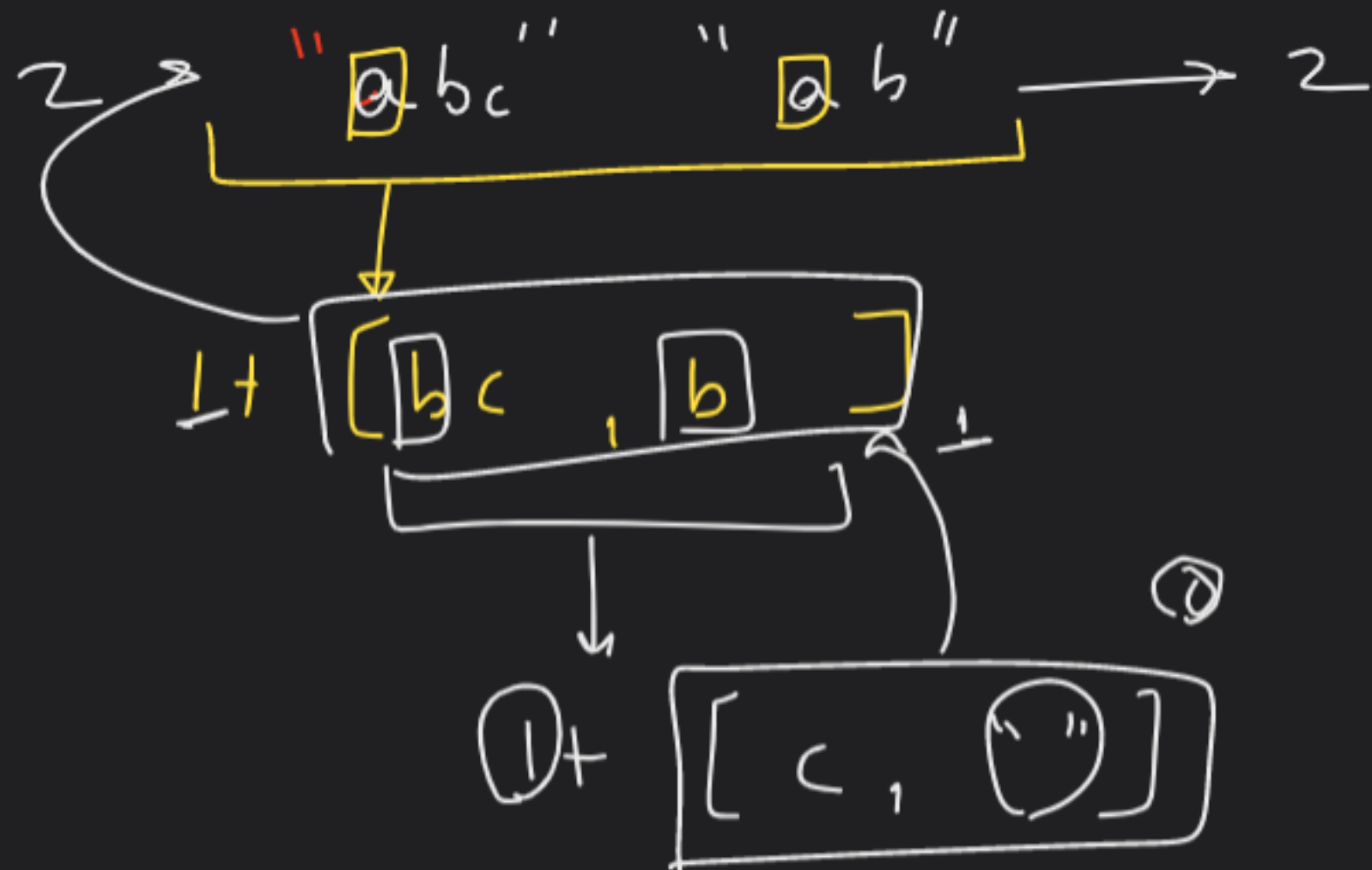
LCS

$\text{solve}(i, j+1)$

$\max(\text{solve}(i, j+1), \text{solve}(i+1, j))$



solve(i+1, j)



(3) → "abcd", "ace"

a

output: ""

1 +

~~$$[ [b]_K, c, d, e, [c]_e ]$$~~

LCS

max / ~ stuff -

~~[b c d e]~~

~~$$[c]_c, [c]_c \leftarrow 2$$~~
$$\mathbb{Q}_+ \left[ \sqrt{d}, \sqrt{e} \right]$$
~~[bcdc, ""]~~
$$[c d, e]$$

$(d_c, \dots)$  ,  $(\dots)$

Yes there



anspt 2<sup>nd</sup>  
max h



# → Longest Arithmetic Subsequence

(A.P)  
Arithmetic  
Progression

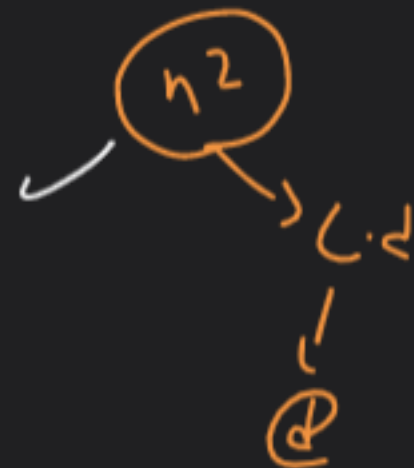
i/p → [9, 4, 7, 2, 10]

# work → find all

A.P sequence

get the longest

1/1



[ — , — , — , — ]

```
for ( int i = 0; i < n; i++ )
```

```
{ for ( int j = i + 1; j < n; j++ )
```

```
{
```

```
}
```



[9, 4, 7, 2, 10]  
i j

A.P

1 3 5 7 9 11 13 15

$$a, a+d, a+2d, a+3d, a+4d, \dots$$

— — — — — ( + d )

2 + (-7)

HARD

# Longest Arithmetic Subsequence

2 min

2 - 7 = -5

7, 2

7, 10

① → 2 no

1



2 - 7 = -5  
2 + (-1) = 1

pari break

7 - 4 = 3  
7 + 3 = 10

4 - 9 = -5

4 + (-1) = -1

~~9 - 7 = 2~~

2 → 2 - 7

7 - 9 = (-2)

2 + (-4) = -2

7 + (-2) = 5

10 - 7 = 3

①

9, 4

② 4, 7, 10

③

②

9, 7

④ 4, 2

③

9, 2

⑤ 4, 10

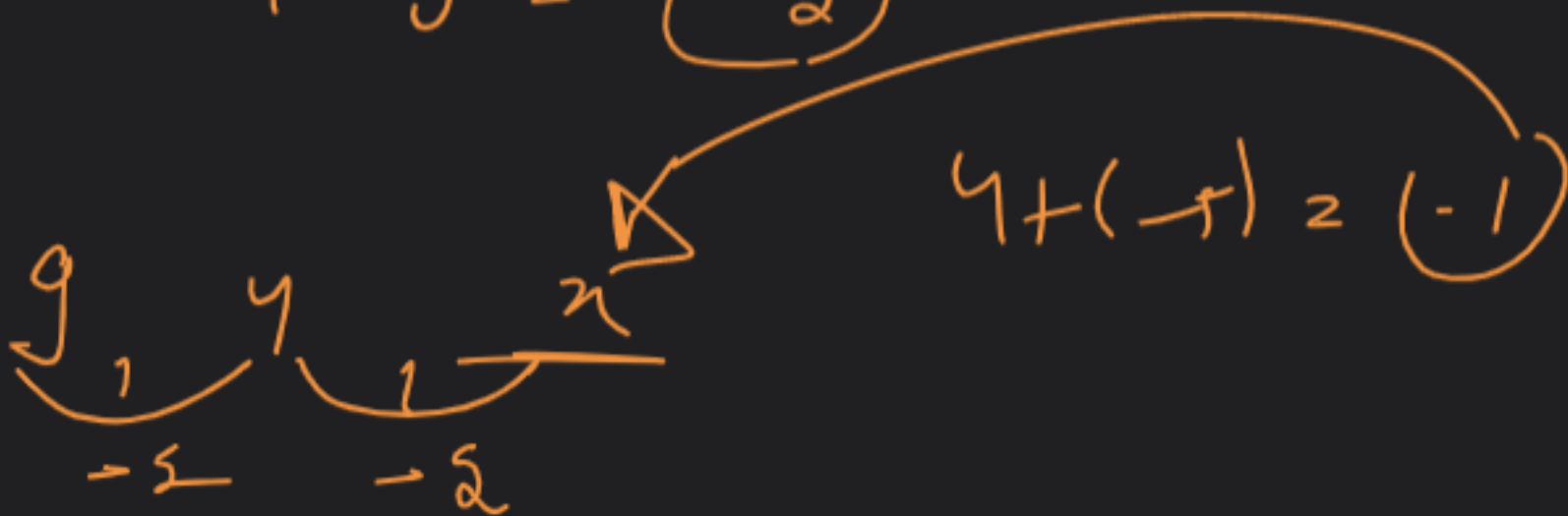
④

9, 10



$$5 + (2) = \textcircled{7}$$

$$4 - 9 = \textcircled{-5}$$



$$4 + (-5) = \textcircled{-1}$$

```
for (int i = 0; i < n; i++)
```

```
{
```

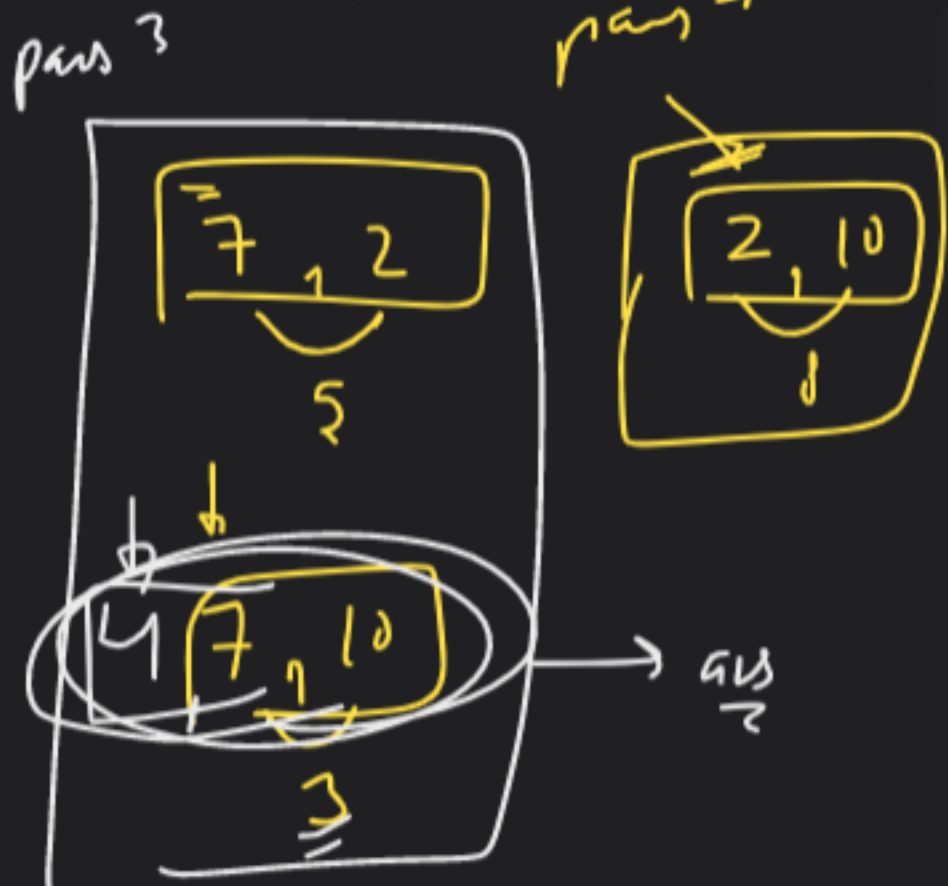
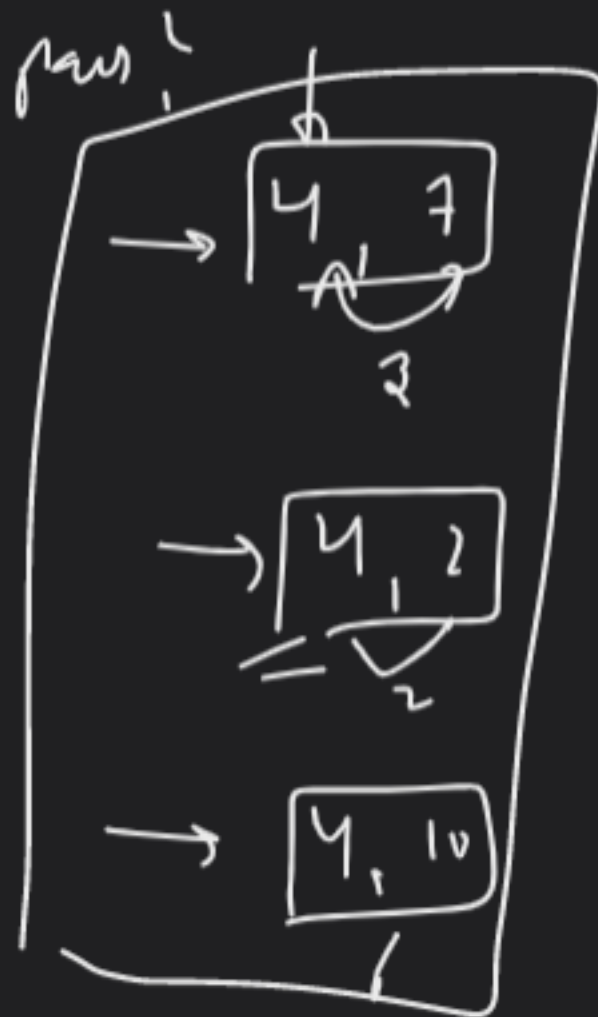
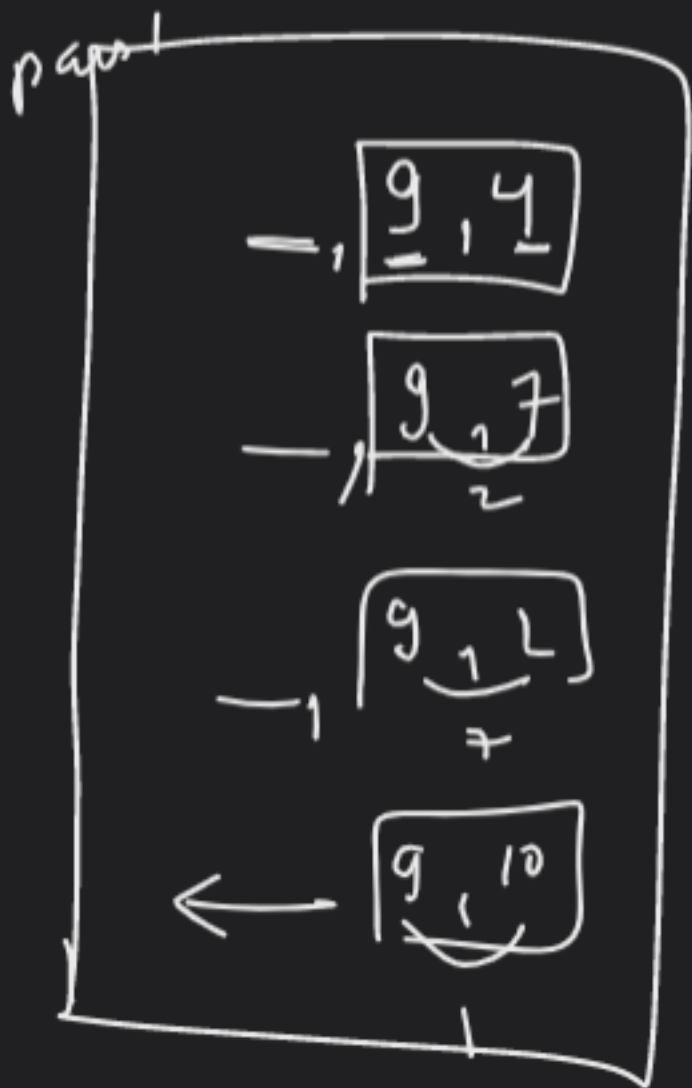
```
    for (int j = i+1; j < n; j++)
```

```
    {
```

```
    }
```



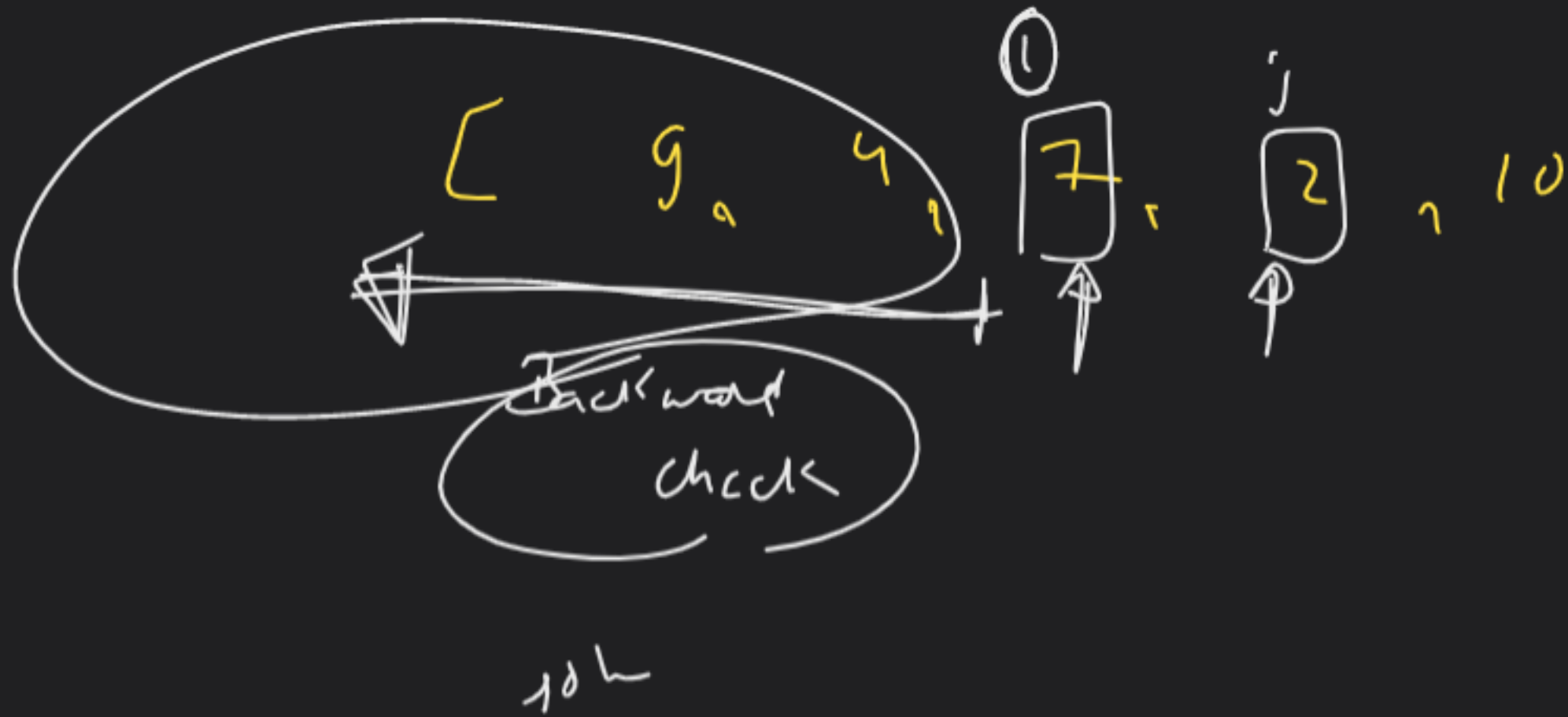
$[9, 4, 7, 2, 10]$

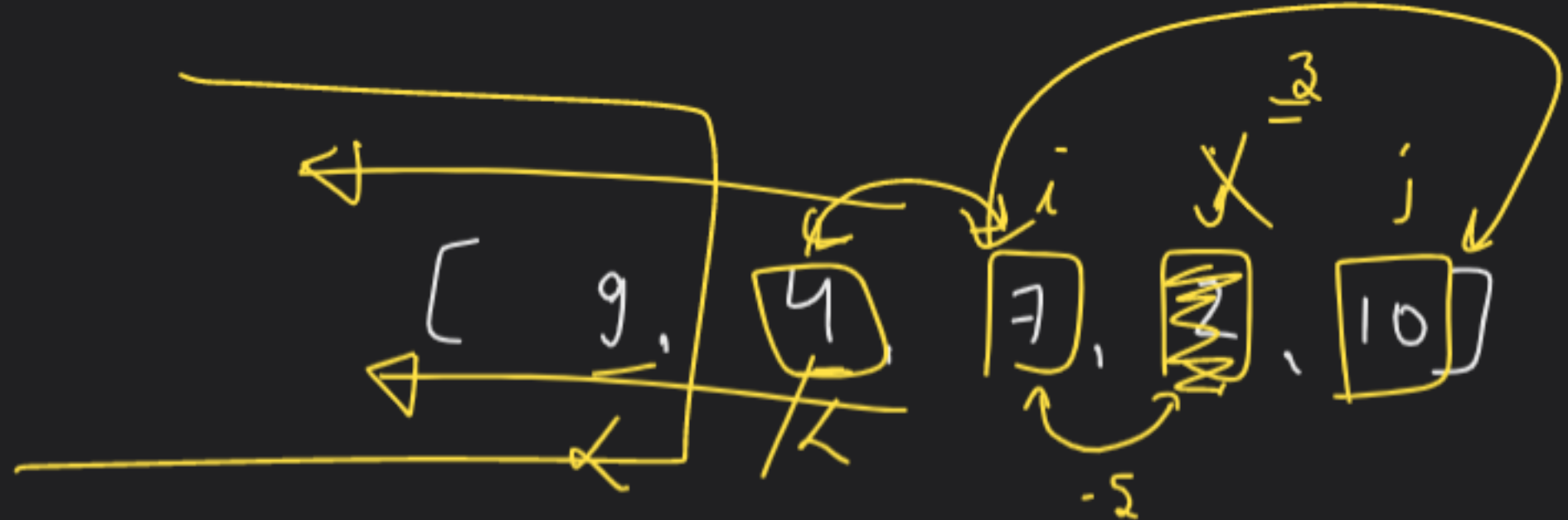


```
int ans = 0;
for (int i = 0; i < n; i++)
{
    for (int j = i + 1; j < n; j++)
    {
        ans = max (ans, 2 + solu(i, j, diff))
    }
}

return ans;
```

$diff = A[j] - A[i]$





solve

1+

(7, 2)

(1 din)

subseq

→ in subseq H/W

→ LIS → B.S  
→ various

→ LCS → various  
LPS → H/W

→ LAS → LAS with diff H/W

→ Distinct Subsequence → H/W