# VIDEO - 31

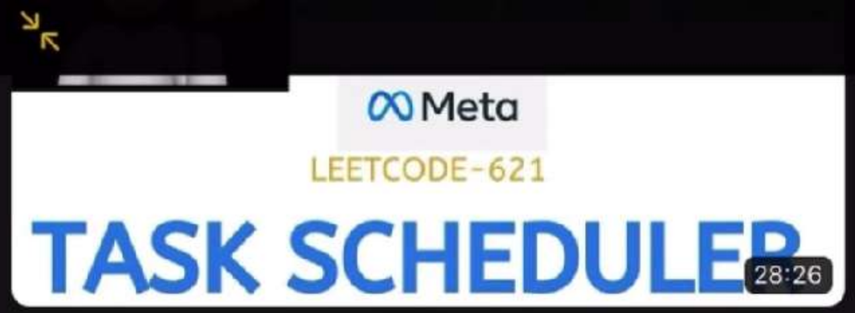
Leetcode
- 621

Medium

Facebook
Instagram → codestorywithMIK

Twith → cswithMIK

→ codestorywithMIK




# 621. Task Scheduler

Medium | Topics | Companies

Meta

You are given an array of CPU `tasks`, each represented by letters A to Z, and a cooling time, `n`. Each cycle or interval allows the completion of one task. Tasks can be completed in any order, but there's a constraint: **identical** tasks must be separated by at least `n` intervals due to cooling time.

Return the *minimum number of intervals* required to complete all tasks.

Example :- { 'A', 'A', 'A', 'B', 'B', 'B' }, n = 2

Output = 8

Example :- { 'A', 'C', 'A', 'B', 'D', 'B' }, n = 1

Output = 6

Example :- { 'A', 'A', 'A', 'B', 'B', 'B' }, n = 2

Output = 8

Example :- { 'A', 'C', 'A', 'B', 'D', 'B' }, n = 1

Output = 6

A B A B C D

Pen

# Thought Process :-

{ 'A', 'C', 'A', 'B', 'D', 'B', 'A', 'A', 'B' } , n=2

A = 4

B = 3

{'A', 'C', 'A', 'B', 'D', 'B', 'A', 'A', 'B'} , n=2

A = 4

B = 3

C = 2

D = 1

{'A', 'C', 'A', 'B', 'D', 'B', 'A', 'A', 'B'}, n=2

A = 4

B = 3

C = 2

D = 1

B = 3

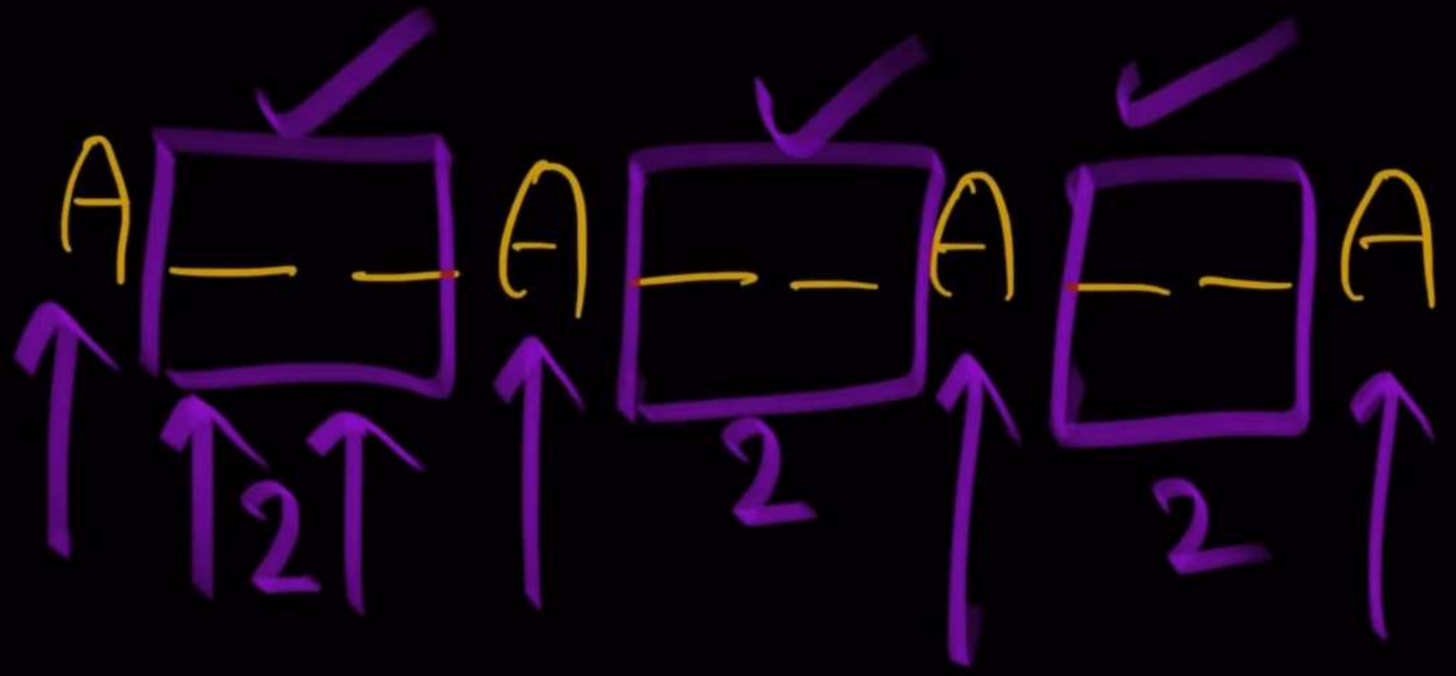
C = 2

D = 1

A _ _ A _ _ A _ _ A

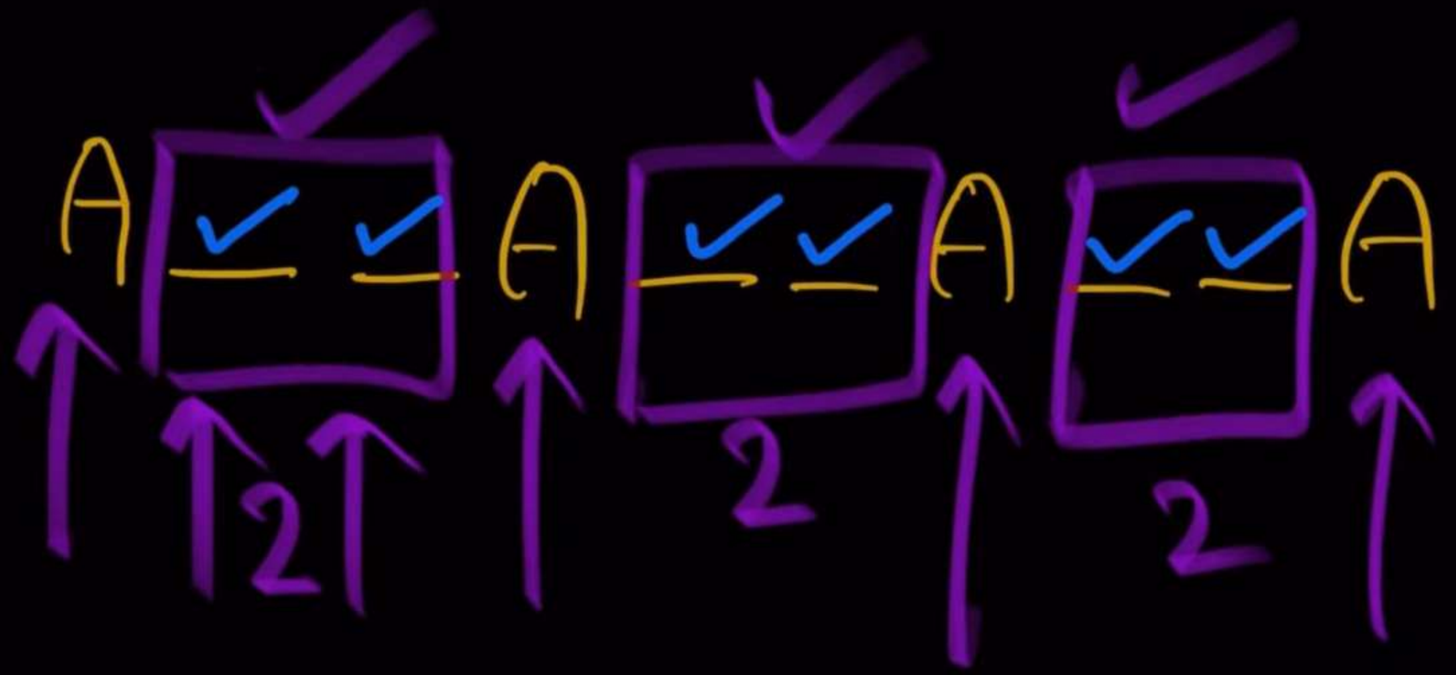
C = 2

D = 1



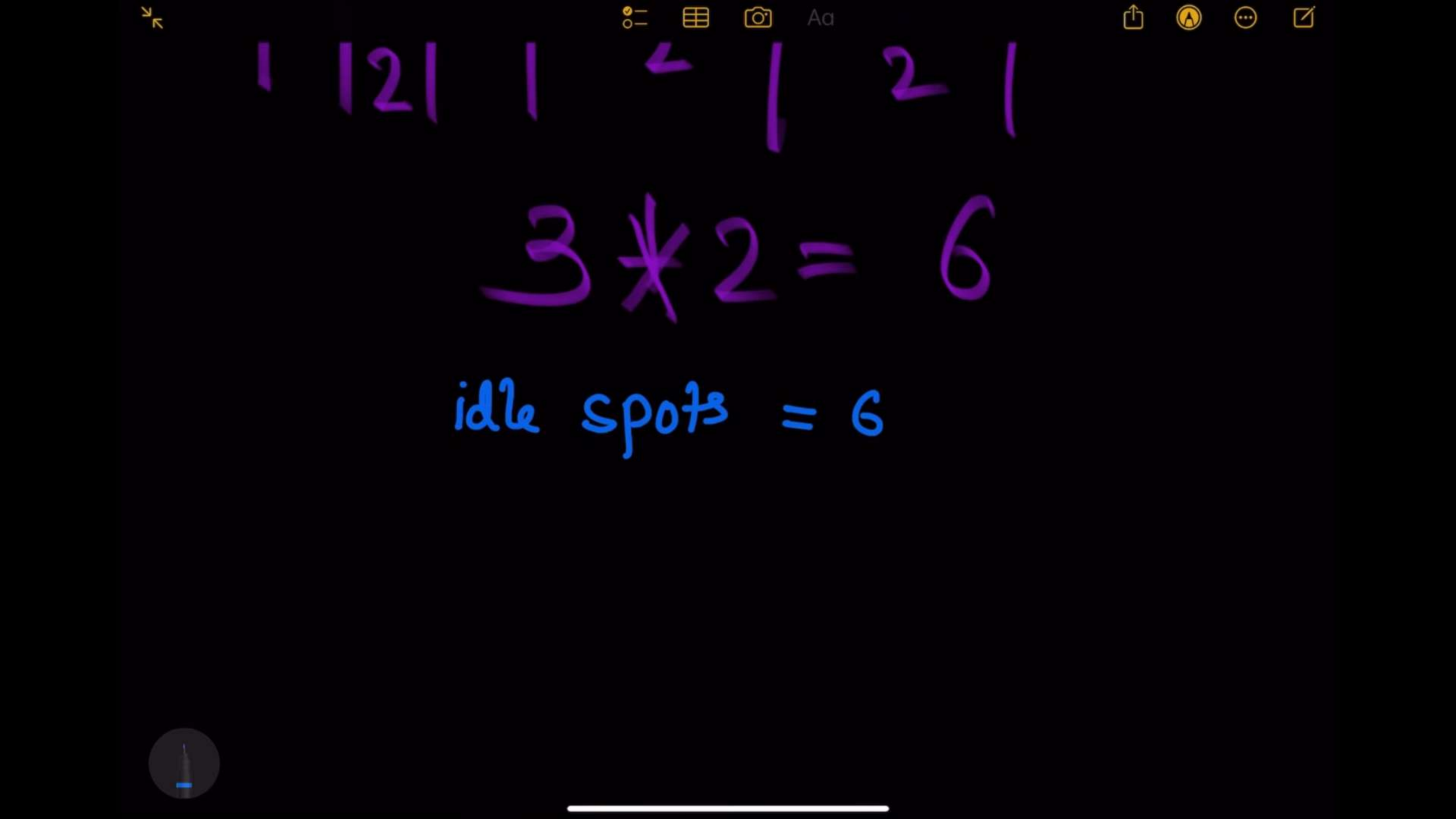
C = 2

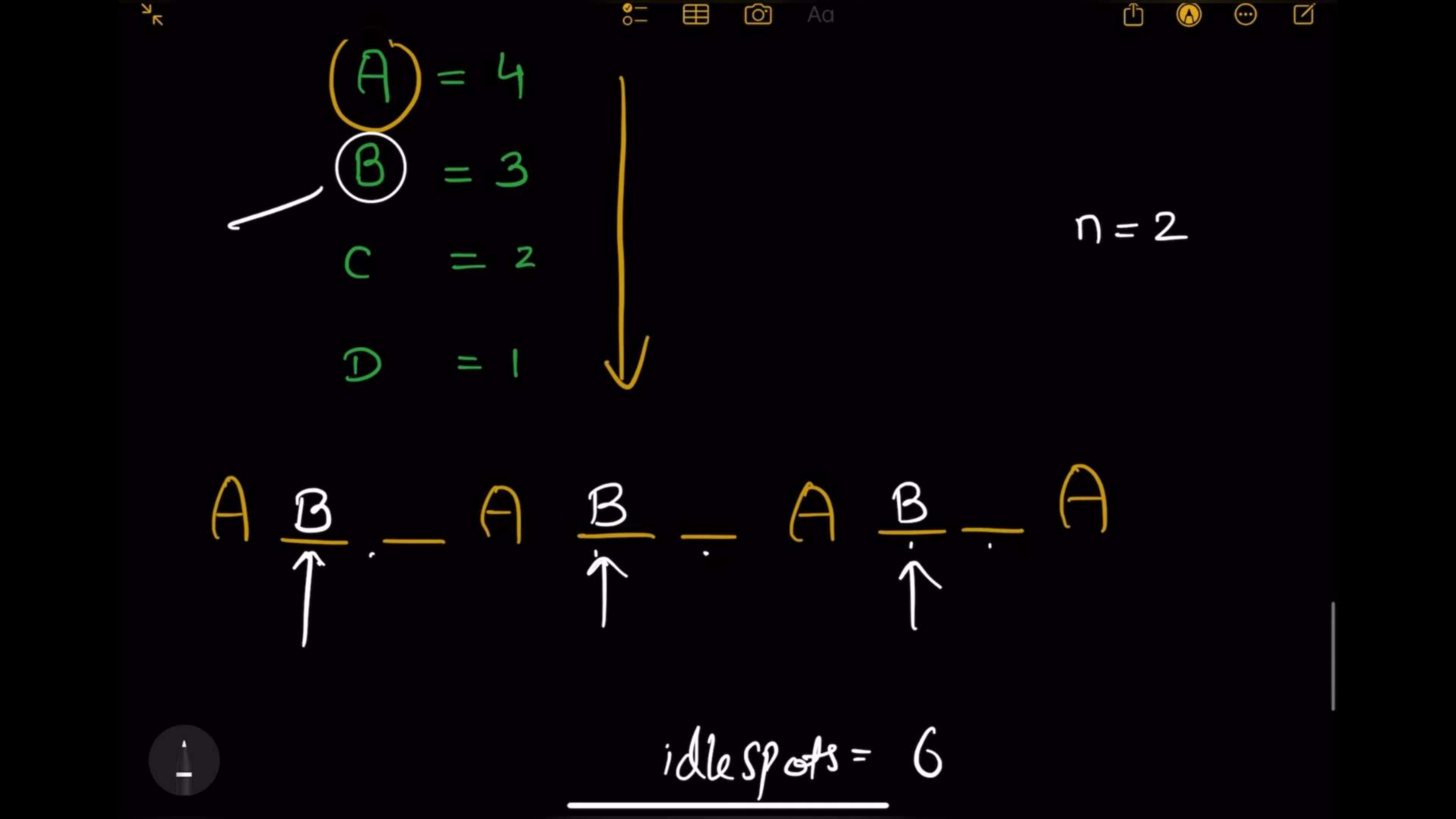
D = 1

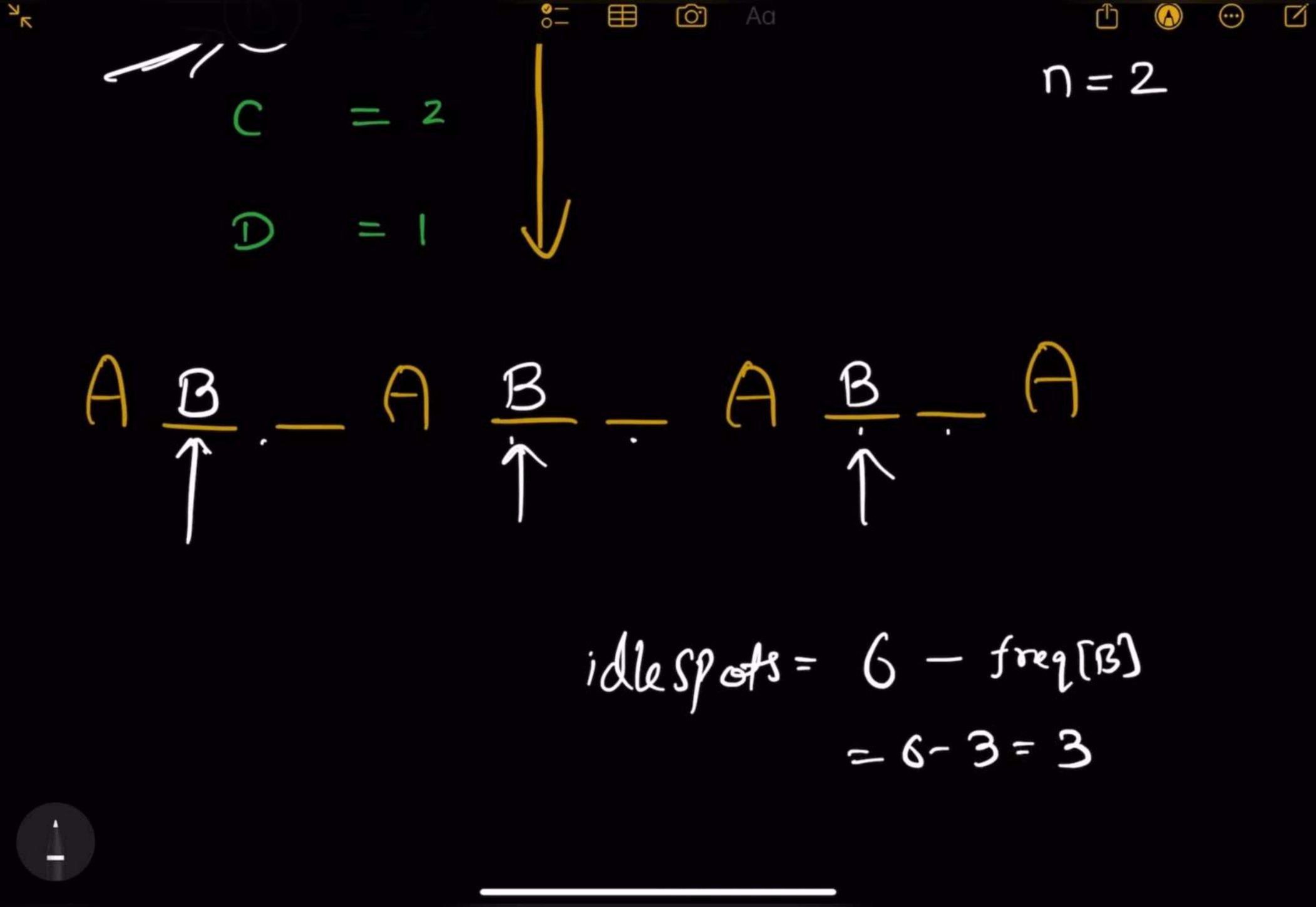



3 * 2 = 6

Monoline


1 | 2 | 1 | 1 | 2 | 1
3 * 2 = 6
idle spots = 6


A = 4
B = 3
C = 2
D = 1
n = 2
A B _ _ A B _ _ A B _ _ A
idle spots = 6


C = 2
D = 1
n = 2
A B _ _ A B _ _ A B _ A
idle spots = 6 - freq[B]
= 6 - 3 = 3

A B C A B C A B _ A

idlespots = 6 − freq[B]

= 6 − 3 = 3

= 3 − fr[C]

= 3 − 2

= 1

A B C A B C A B _ A

idlespots = 6 − freq[B]

= 6 − 3 = 3

= 3 − fr[C]

= 3 − 2

= 1

A B C A B C A B D A

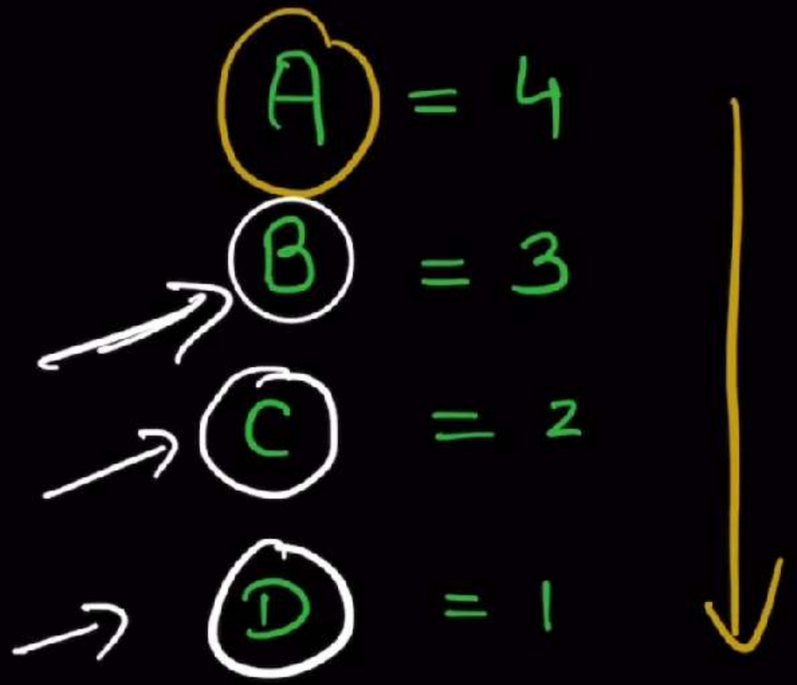
idleSpots = 6 − freq[B]

= 6 − 3 = 3

= 3 − fr[C]

= 3 − 2

= 1 − 1 = 0




n = 2

A B C A B C A B D A

n = 10

idleSpots = 6 − freq[B]

{'A', 'C', 'A', 'B', 'D', 'B', 'A', 'A', 'B', 'B'}, n = 2

A = 4

B = 4

C = 1

D = 1

A _ _ A _ _ A _ _ A

A = 4

B = 4

C = 1

D = 1

A _ _ A _ _ A _ _ A

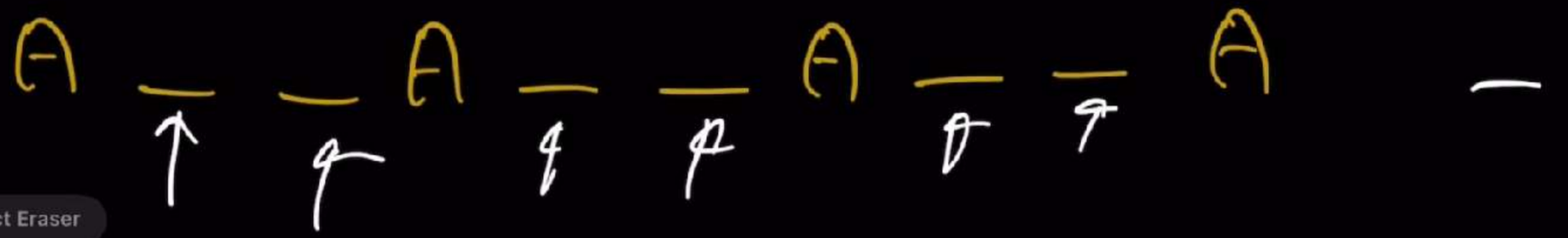
idlespots = G

$A = 4$

$B = 4$

$C = 1$

$D = 1$



idlespots = G

A = 4

B = 4 1

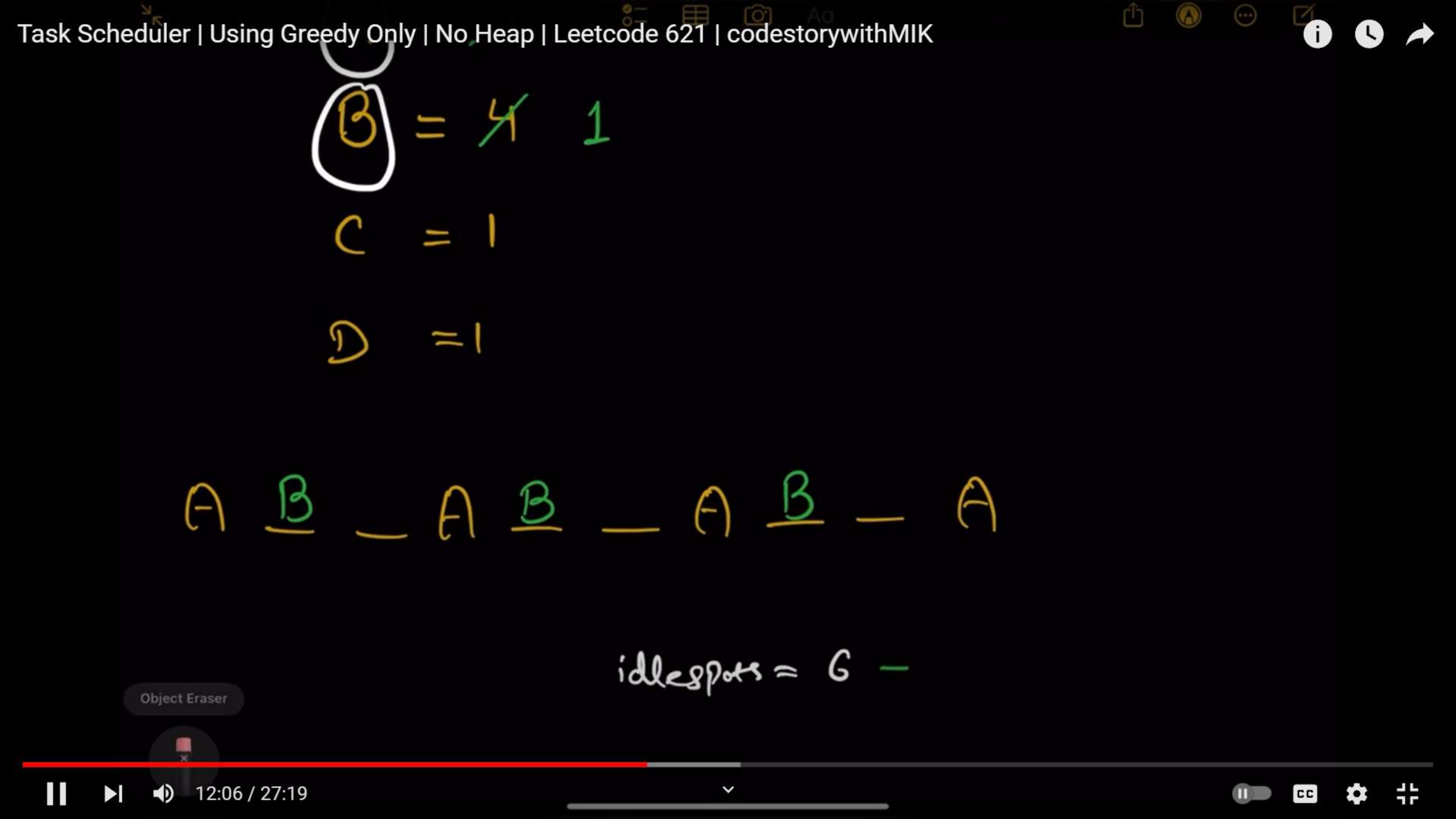
C = 1

D = 1

A B _ A B _ A B _ A

idlespots = 6

Task Scheduler | Using Greedy Only | No Heap | Leetcode 621 | codestorywithMIK



(D) = ↗ 1

C = 1

D = 1

Gaddha = 3

A B _ A B _ A B _ A

idlespots = G − Gaddha

= 6 − 3

= 3

A = 4 0

B = 4 1 ☆

C = 1

D = 1

Gaddha = 3

A B _ A B _ A B _ A

idlespots = 6 − Gaddha

= 6 − 3

C = 0

D = 1

Gaddha = 3

A B C A B _ A B _ A

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c]

B = 4 1 ☆

C = X O

D = 1

Gaddha = 3

A B C A B _ A B _ A

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c]

3 − 1

Gaddha = 3

A B C A B _ A B _ A

idlespots = G − Gaddha

= 6 − 3

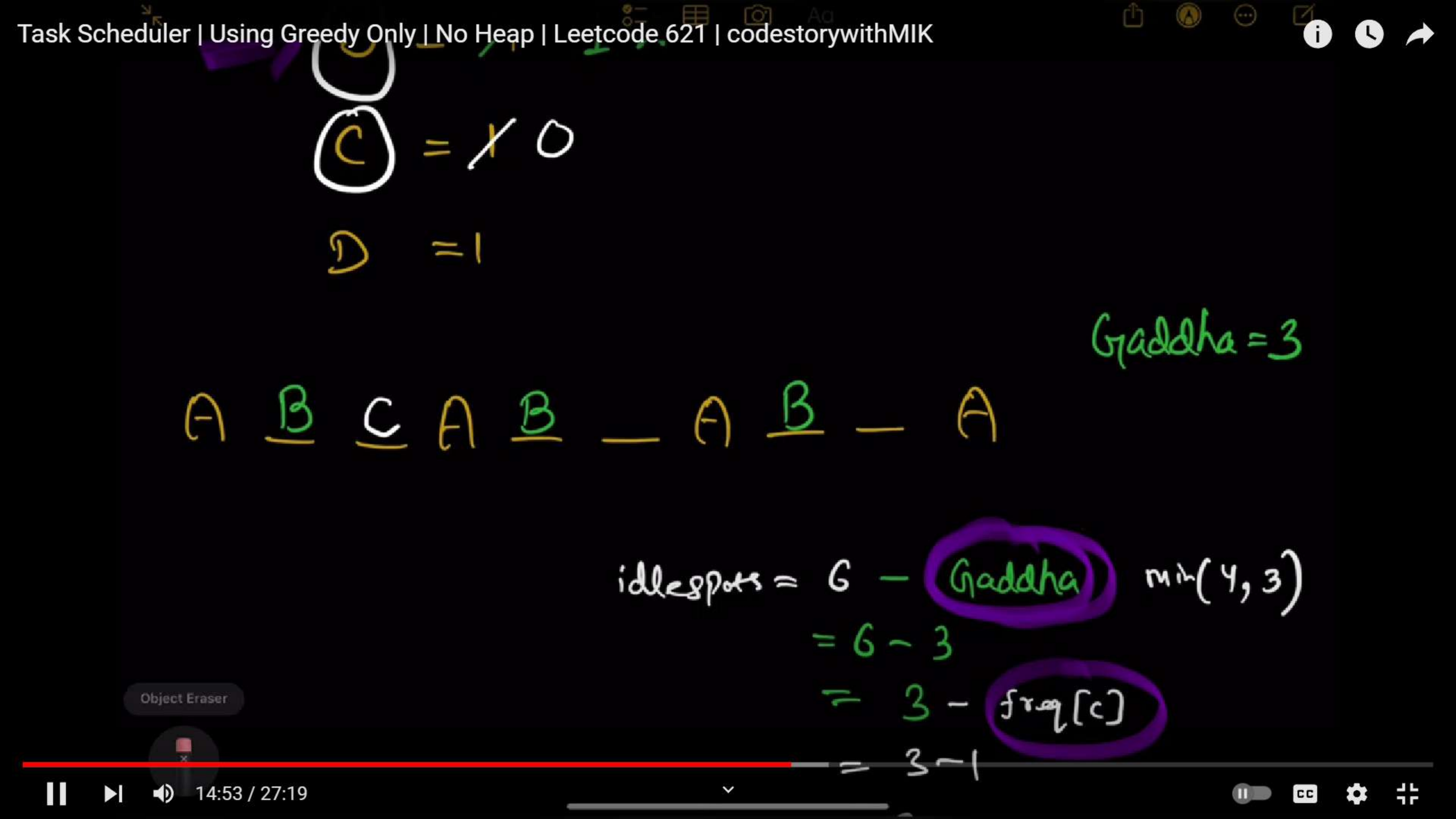
= 3 − freq[c]

= 3 − 1

= 2

min( freq[ch] , Gaddha )

Task Scheduler | Using Greedy Only | No Heap | Leetcode 621 | codestorywithMIK



idlespots = G − Gaddha

= 6 − 3

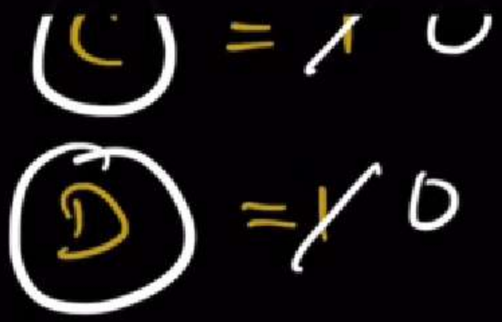
= 3 − freq[c]

= 3 − 1

= 2

$$\frac{}{\min(\text{freq}[ch], \text{Gaddha})}$$

Pen




Gaddha = 3

A B C A B D A B _ A

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c] min(1, 3)

= 3 − 1

= 2 − min(fr[D], 3)

= 2 − min(1, 3)

A B C A B D A B _ A

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c] min(1, 3)

= 3 − 1

= 2 − min(fr[D], 3)

= 2 − min(1, 3)

= 2 − 1 =

min(freq[ch], Gaddha)

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c] min(1, 3)

= 3 − 1

= 2 − min(fr[D], 3)

= 2 − min(1, 3)

= 2 − 1 = 1

min(freq[ch], Gaddha)

C = 10

D = 10

n = 10

Gaddha = 3

| A | B | C | A | B | D | A | B | _ | A | B |
|---|---|---|---|---|---|---|---|---|---|---|

n + idle spots

idlespots = G − Gaddha

= 6 − 3

= 3 − freq[c] min(1, 3)

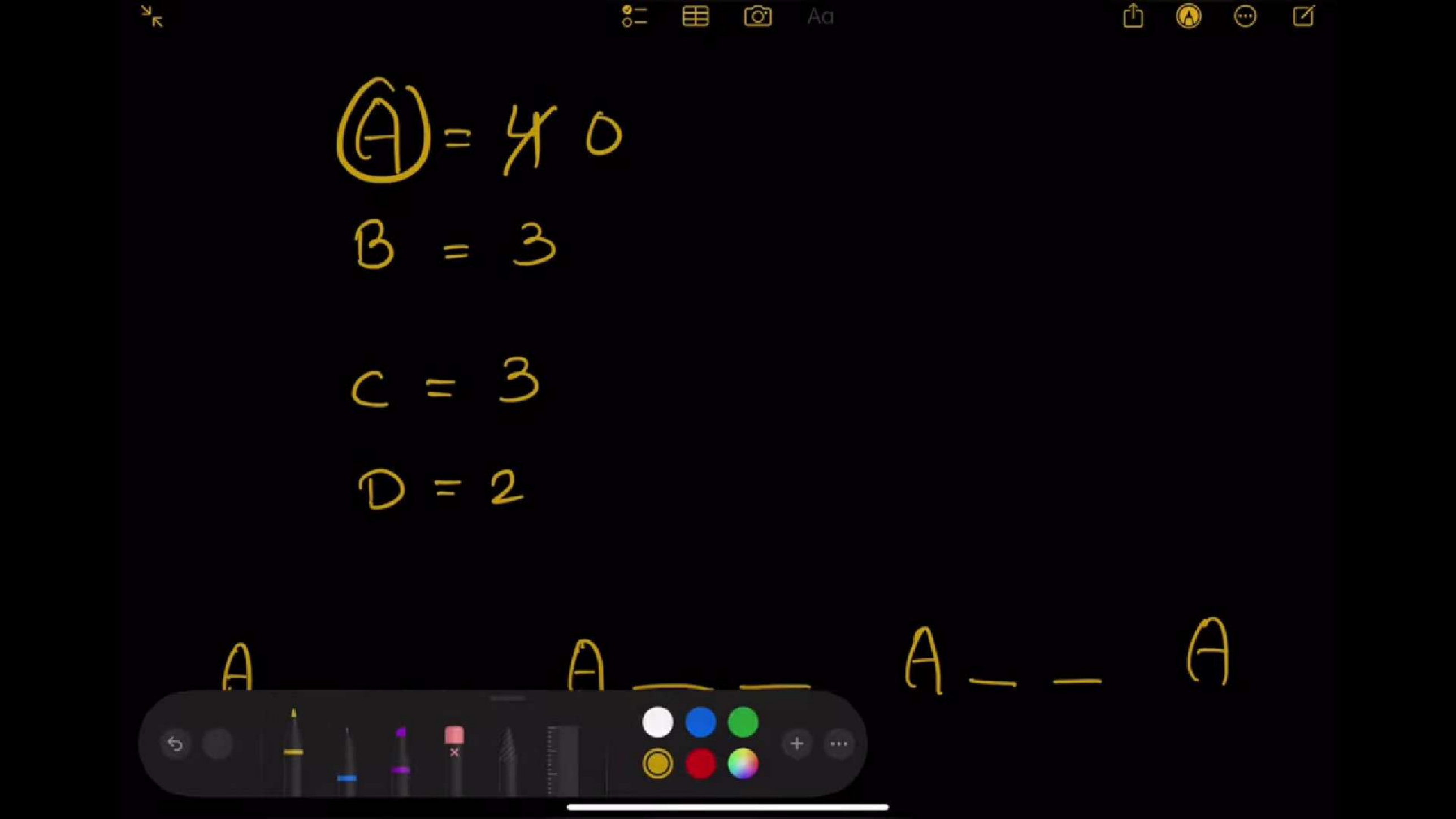
= 3 − 1

A = 4

B = 3

C = 3

D = 2


A = 40
B = 3
C = 3
D = 2
A
A
A
A

A = 40

B = 3

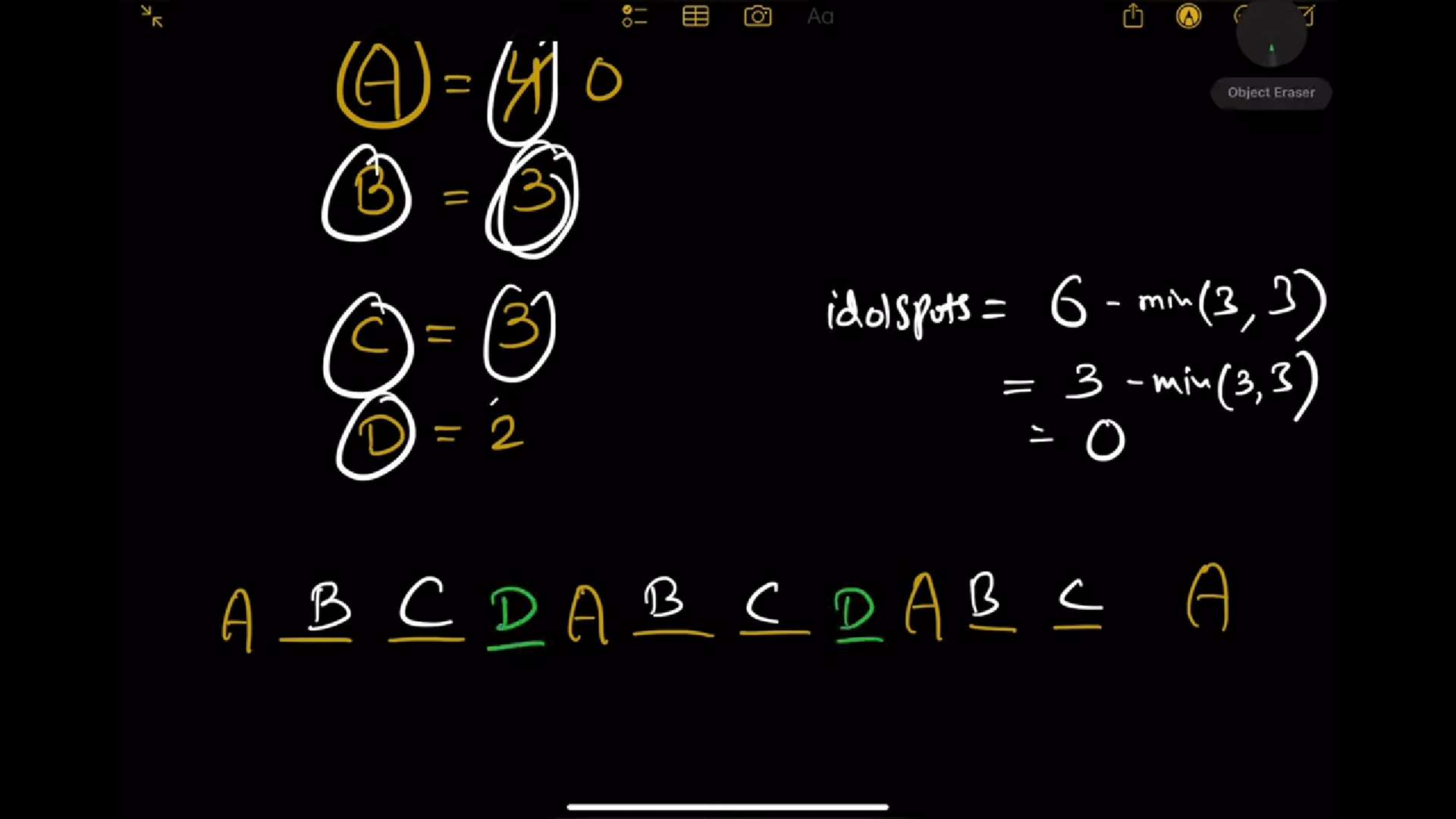
C = 3

D = 2

idolSpots = 6 - min(3, 3)

= 3 - min(3,

A B _ _ A B _ _ A B _ A


A = 4
B = 3
C = 3
D = 2
idolSpots = 6 - min(3, 3)
= 3 - min(3, 3)
= 0
A B C D A B C D A B C A
Object Eraser

```
if(idolSpots == 0)
    return task.size();

else
    task.size() + idolSpots;
```