CS & IT ENGINEERING

Data Structure

Stack and Queues
Chapter- 4
Lec- 06

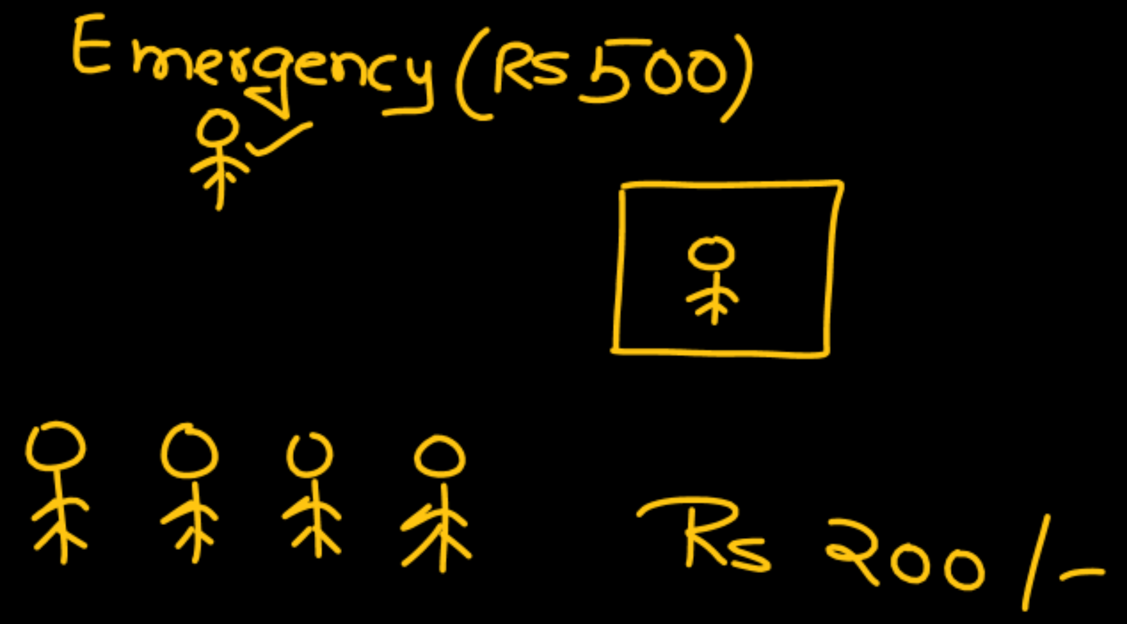
By- Pankaj Sharma sir

TOPICS TO BE COVERED

- Queue-II

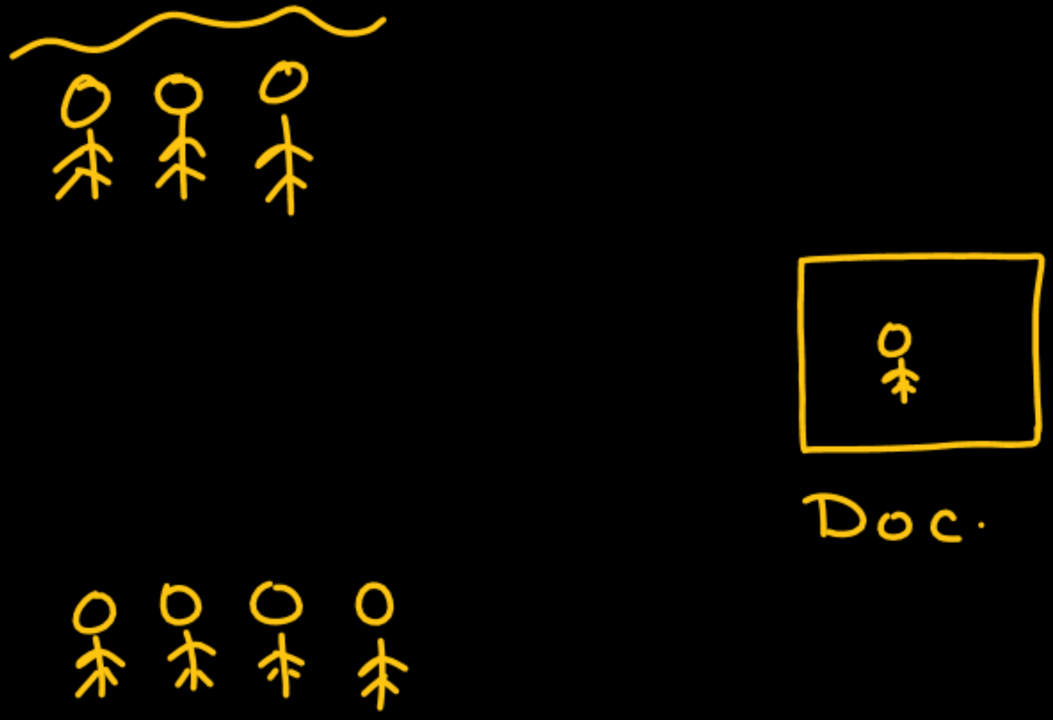
Priority Queue

A priority is associated with each element.



Element with high priority processed first.

Same priority

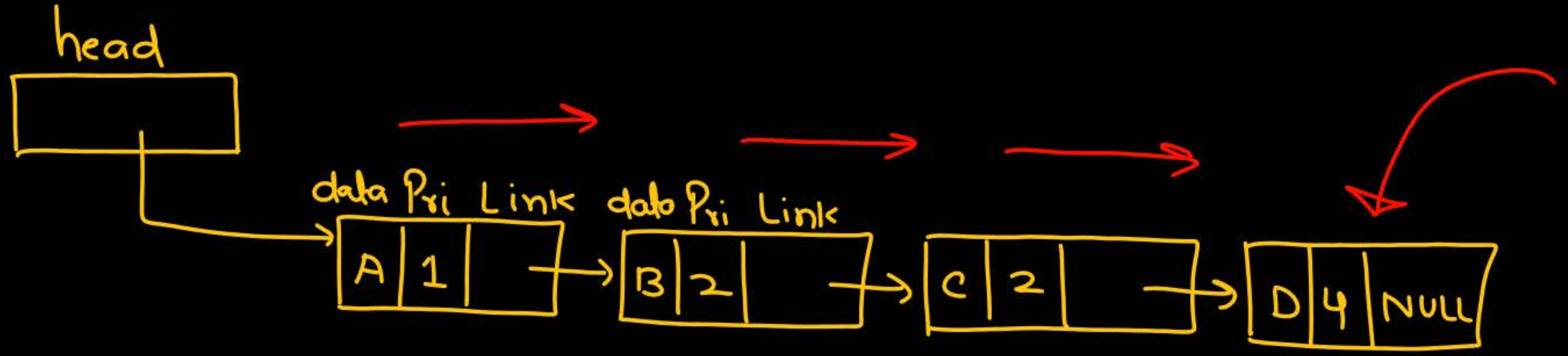


Elements with same priority are pro. as per their arrival.

Small no → High Priority ✓ gnge

Large no — High Priority

(E, 2)



B पहले आया है |

A Priority Queue Q is used to implement a stack S that stores characters. Push(C) is implemented as Insert(Q, C, K), where K is an app. integer key choosen by implementation.

Pop is implemented as DELETEMIN(Q). For a seq. of operations the key choosen are in :

A) Non-increasing order.

B) Non-decreasing order.

C) strictly-increasing order

D) strictly-dec. order

$\xrightarrow{1,2,3}$ inc. X

$\xrightarrow{A,B,C}$

Stack ⇒ using a Priority Queue Q

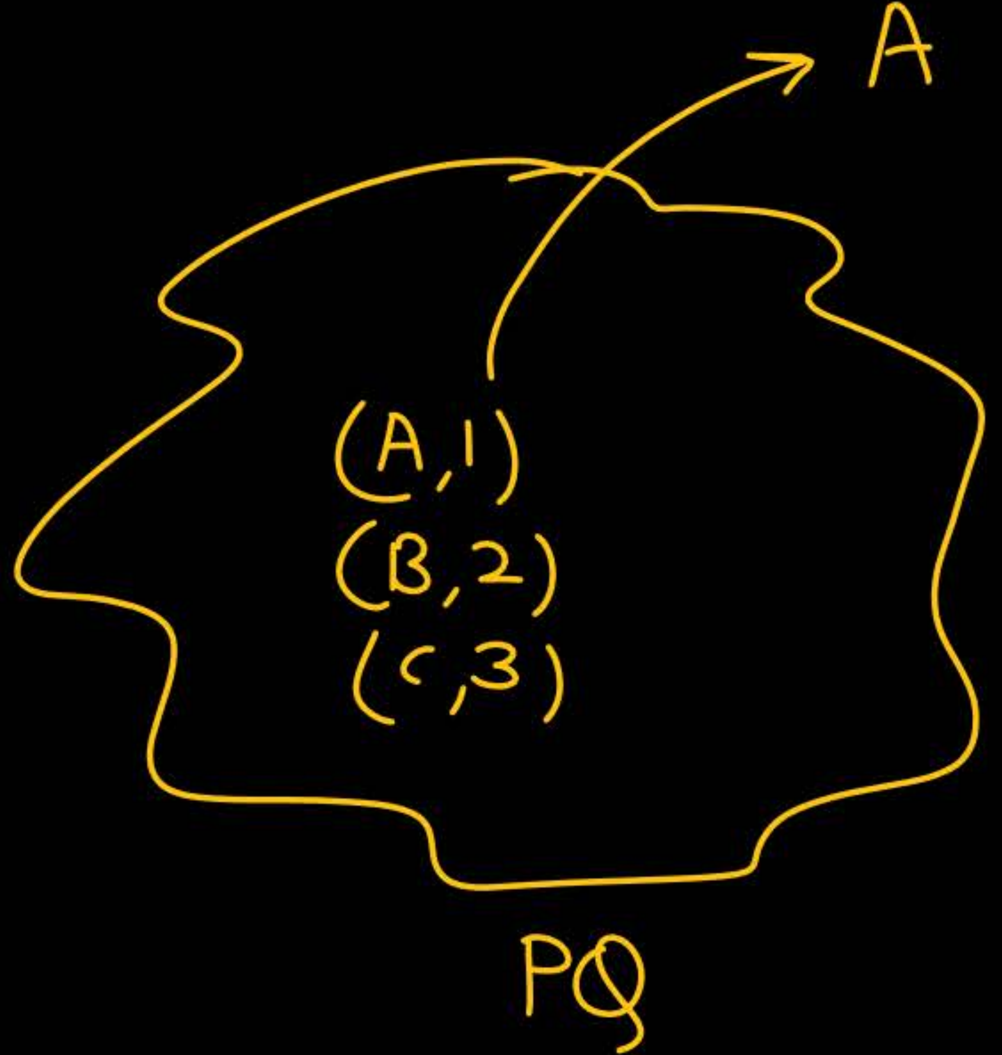
Push('A') ⇒ Insert(Q, 'A', 1) → key

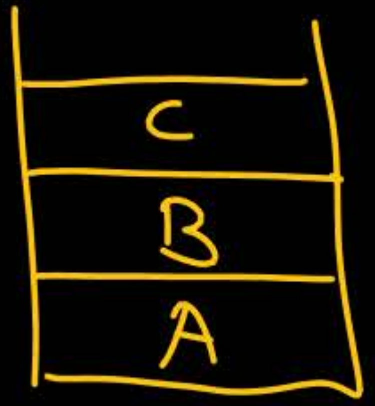
Push('B') ⇒ Insert(Q, 'B', 2) → key

Push('C') ⇒ Insert(Q, 'C', 3)

X

Pop ⇒ DeleteMin(Q)



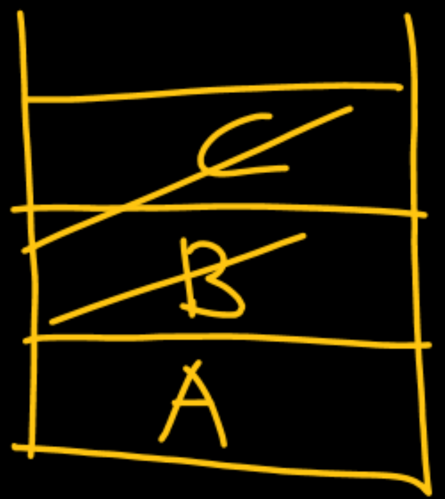


Pop() ⇒ C

Push('A') ⇒ Insert(Q, 'A', 3)

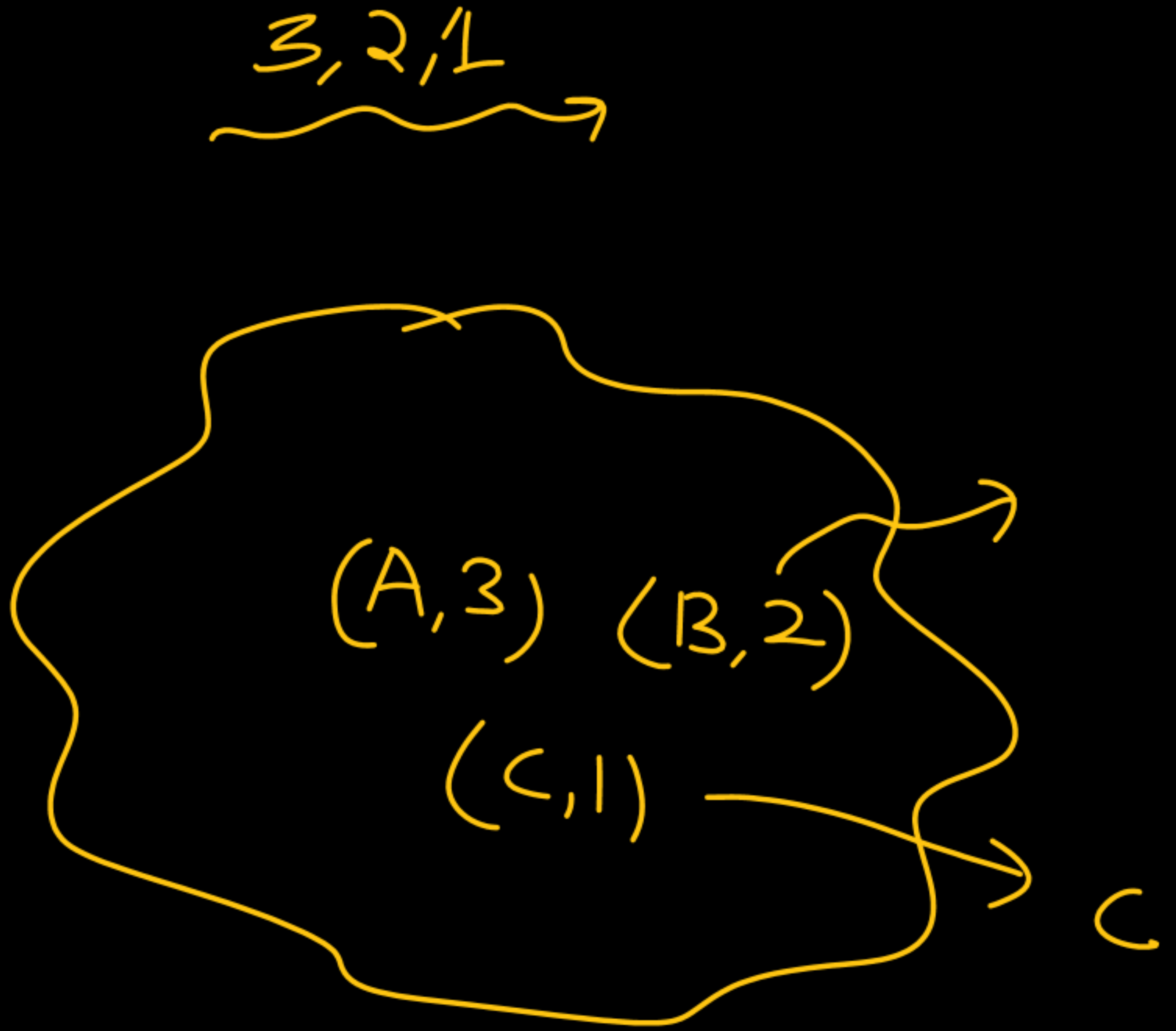
Push('B') ⇒ Insert(Q, 'B', 2)

Push('C') ⇒ Insert(Q, 'C', 1)



Pop() ⇒ C

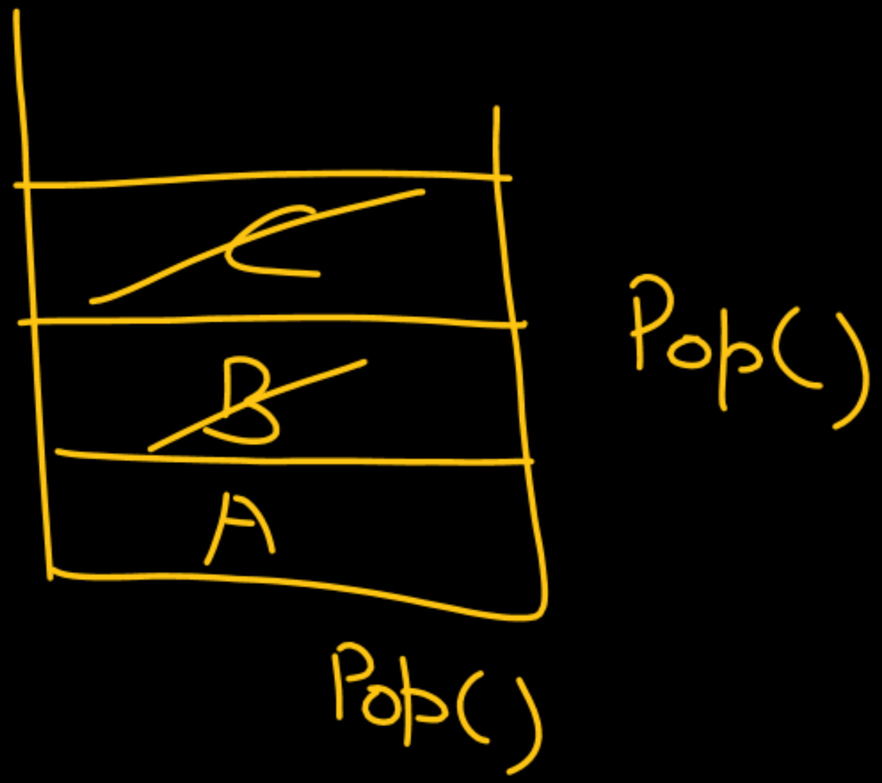
Pop() ⇒ B

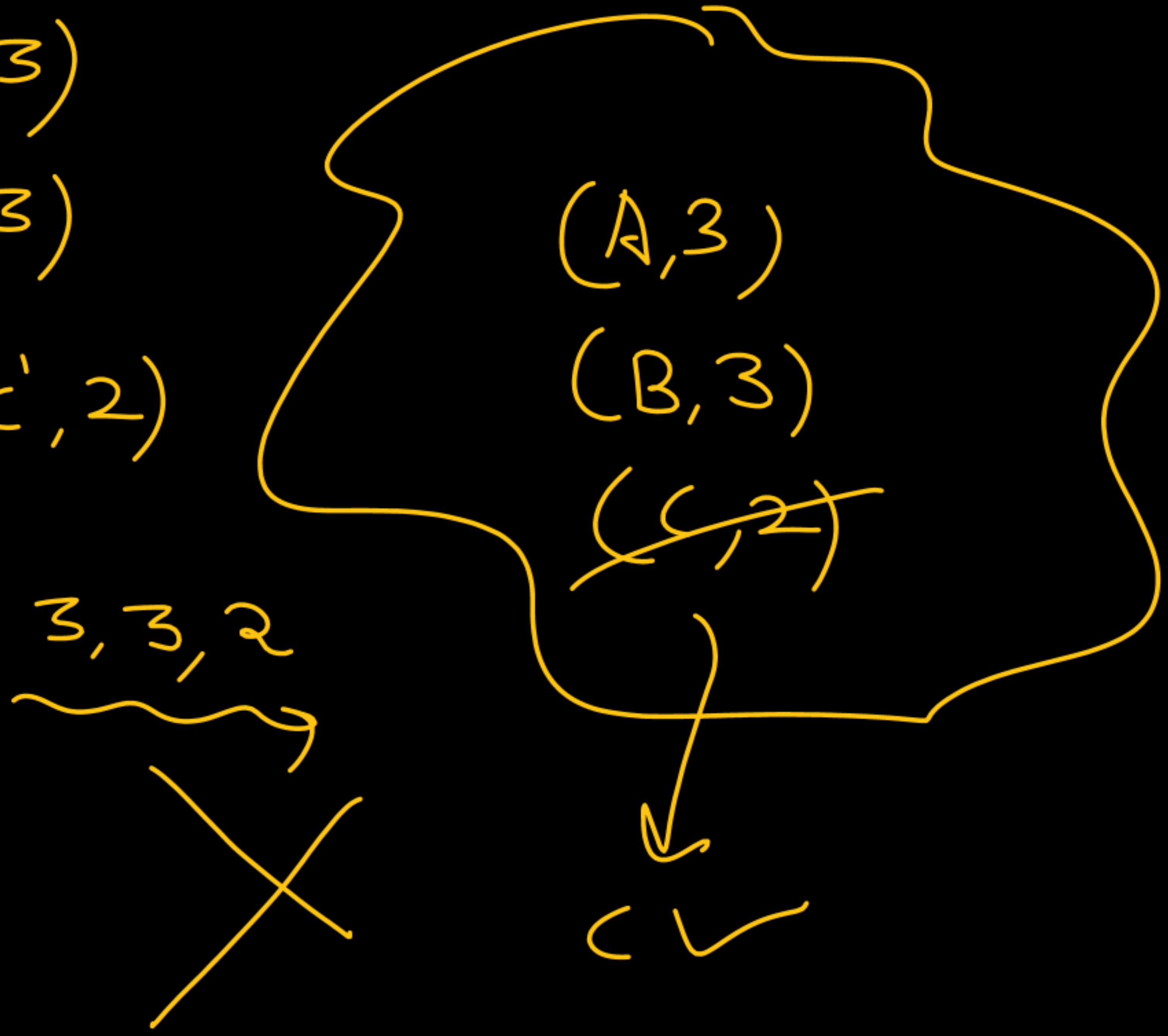


Push('A') ⇒ Insert(Q, 'A', 3)

Push('B') ⇒ Insert(Q, 'B', 3)

Push('C') ⇒ Insert(Q, 'C', 2)





Q Gate-2001

What is the min. no. of {stacks of size n} required to implement a queue of size n?

A) 1

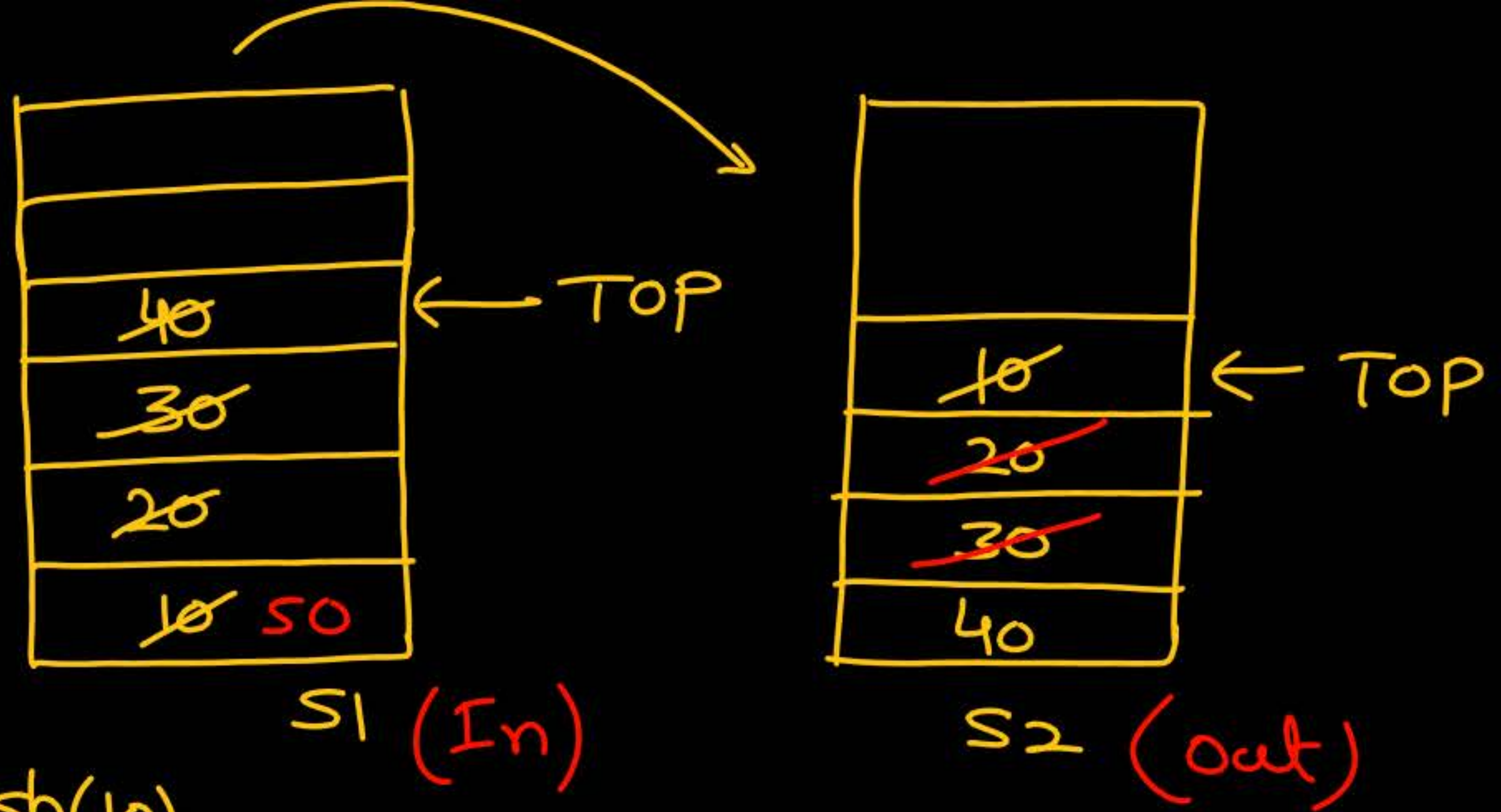
B) 2

C) 3

D) 4

Q

~~10~~	~~20~~	~~30~~	40	50	



Insert(10)

Insert(20)

Insert(30)

Insert(40) ⟶ ~~10~~, 20, 30, 40

delete( ) ~~20~~, 30, 40

→ delete( ) 30, 40

Insert(50) 30, 40, 50

delete 40, 50

Push(10)

Push(20)

Push(30)

Push(40)

delete() ⇒ (i) Move all the elem. from S1 to S2

(ii) Pop() from S2

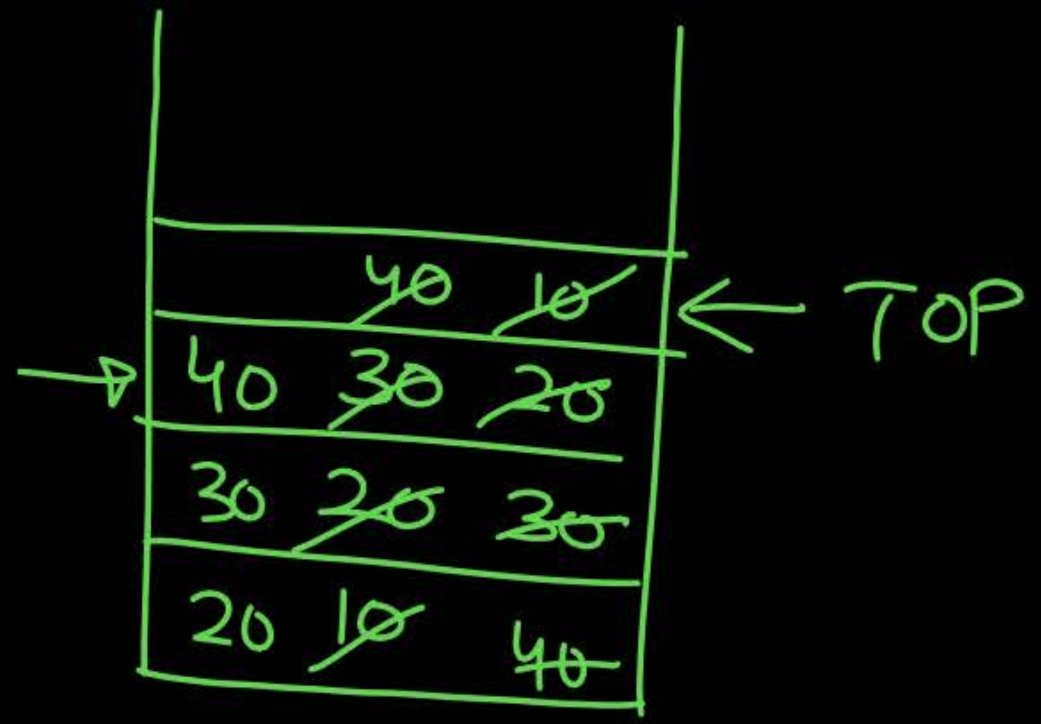
delete ⇒ look at S2, if it is not Empty

Magical stack → REVERSE op. Support.

Insert → Push( )

~~10~~	20	30	40

delete 20 30 40



⇒ (i) REVERSE
(ii) Pop( )
(iii) REVERSE

Gate 2003

help

Let S be a stack of size $\boxed{n \geq 1}$. Starting with empty stack, suppose we push first n natural no. in sequence and then performs n pop operations. Assume push & pop takes X sec. each.

Y sec. elapse b/w the end of one such stack operation and the start of next operation.

For $m \geq 1$, define the stack life of m as the time elapsed from the end of push(m) to the start of pop operation that remove m from S. The avg. stack life of an ele. in stack is:

A] $n(X+Y)$ ✗

B] $3Y + 2X$ ✗

✓ C] $\boxed{n(X+Y) - X}$

D] $Y + 2X$ ✗

$2(X+Y) - X = X + 2Y$

Push(1)

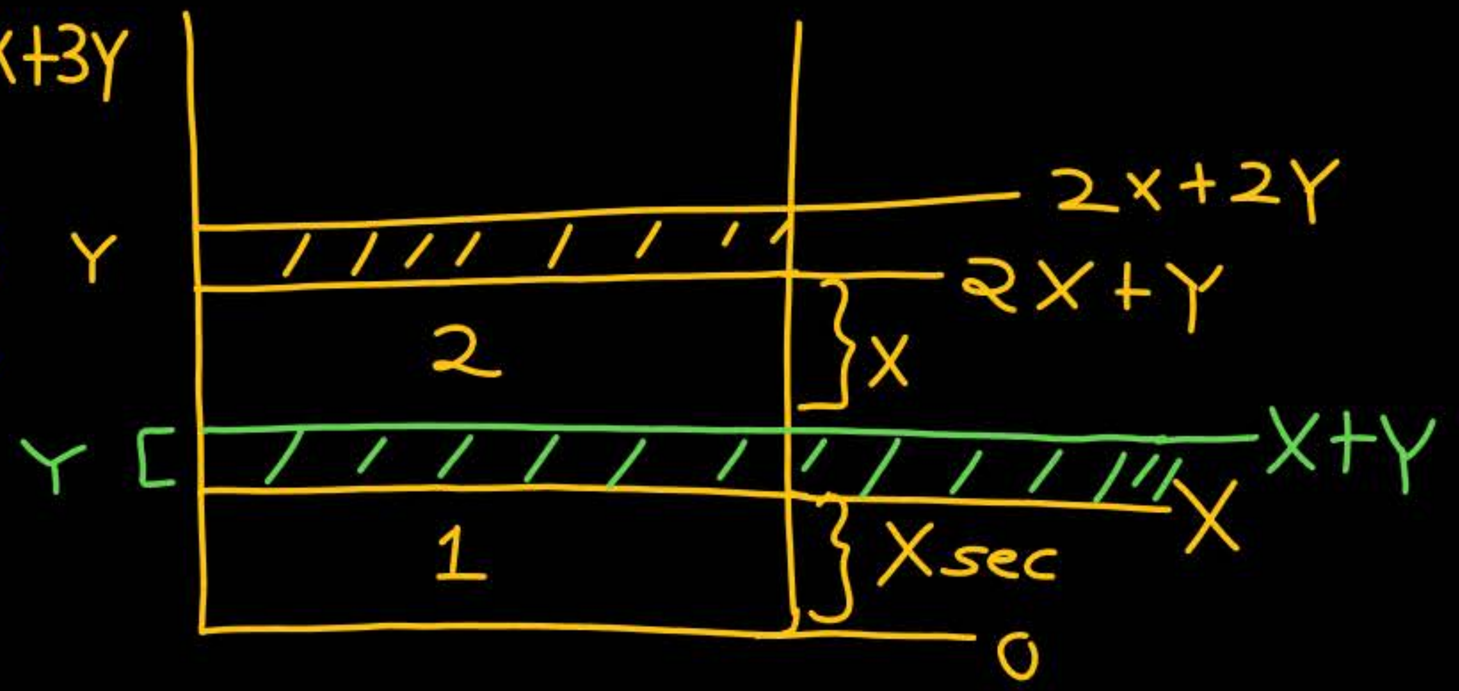
Push(2)

Pop( )

Pop( )

$\boxed{n=2}$

	start Push	End Push	Pop start	Pop End	
1	0	X	3x+3Y	4X+3Y	
2	X+Y	2X+Y	2X+2Y	3X + 2Y	Y

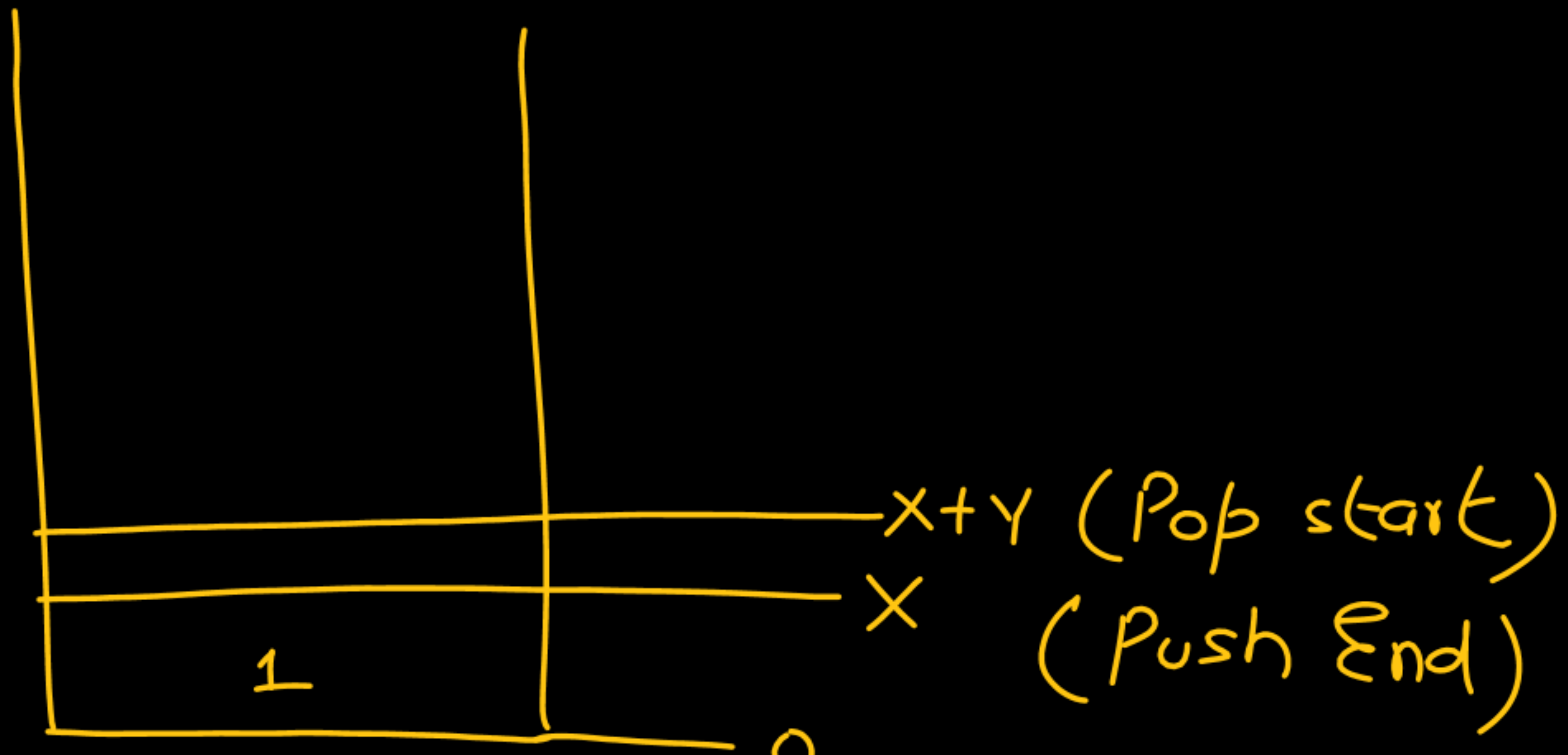


Life time = Pop start - Push end

$\text{Lifetime}_1 = 2X+3Y$

$\text{Lifetime}_2 = Y$

$$\text{avg} = \frac{2X+3Y+Y}{2} = X+2Y$$



$\text{Life time}_1 = (X+Y)-X$

$n=1$

$(X+Y)-X$

Gate 2004

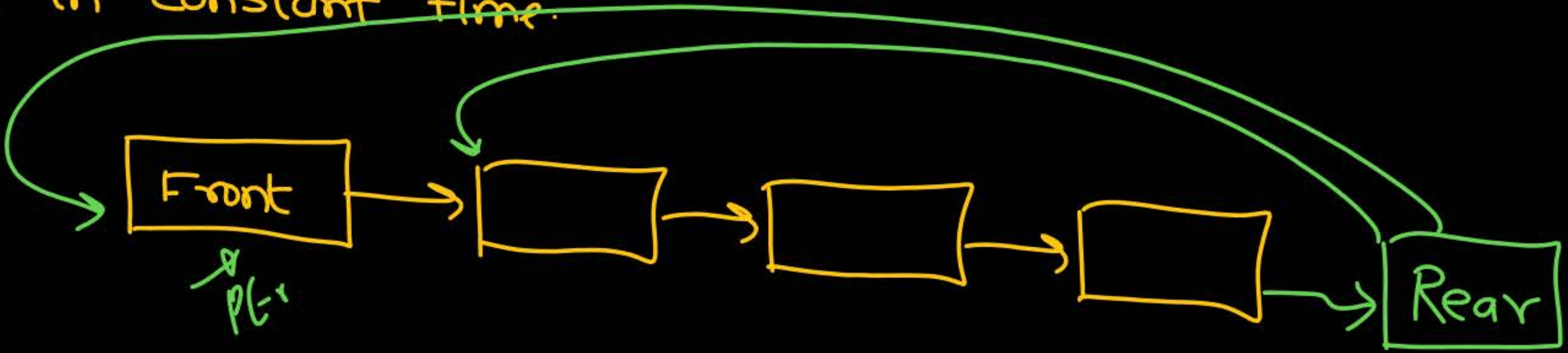
A CLL is used to implement a Queue. A single variable P is used to access the queue. To which node should P point such that both enQueue & deQueue can be Performed in constant time.

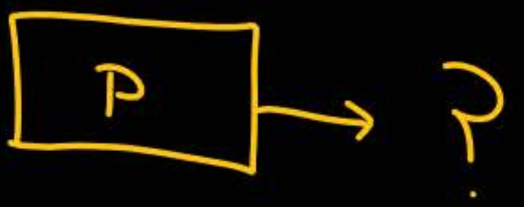
A) rear node

B) front node

C) Not possible

D) Node next to front





Maha-revision

7 PM

C Revise

DS

Tree

Expression trees

THANK YOU GW SOLDIERS !