

Problem 1:

Reverse linked list (Data iterative) :-

(try):

Head ↓

tail ↑

0 → 0 → 0 → 0 → 0 → 0

while (head < tail):

temp = head.data

head.data = tail.data

tail.data = temp

head ++

tail --

LOGIC ↗

} swap

??

A[] | 3 4 8 7
idx 0 1 2 3 4

i i i j j

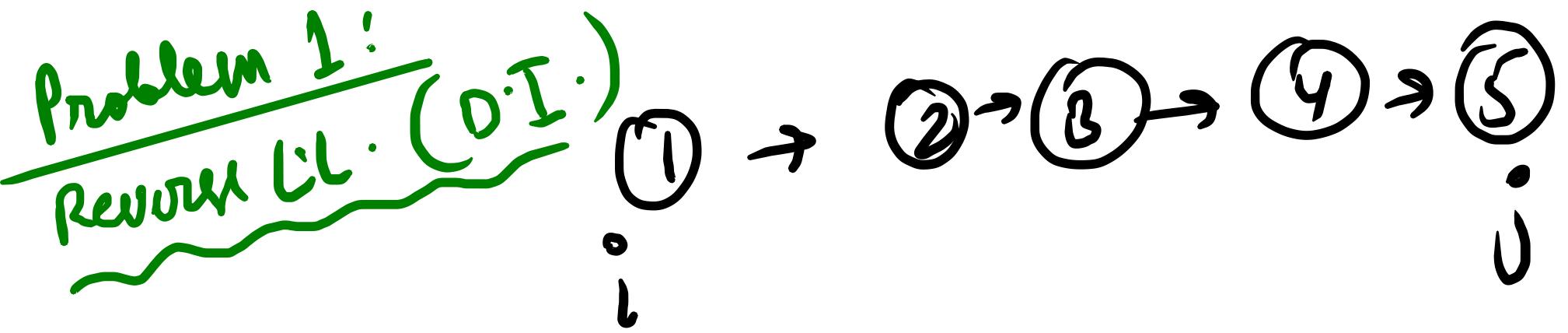
for LL: we: ↗
get NodeAt()

fn ↗

12th Jan, 2022
Lecture - 036

(Hint: Just like array)
Reversal.

=



85:
 $i = 0$
 $j = \text{size} - 1$

① $\text{get Node At } (i)$ ✓
 ② $\text{get Node At } (j)$ ✓

\uparrow

f^n discussed @ prev. class.
 $=$

swap!

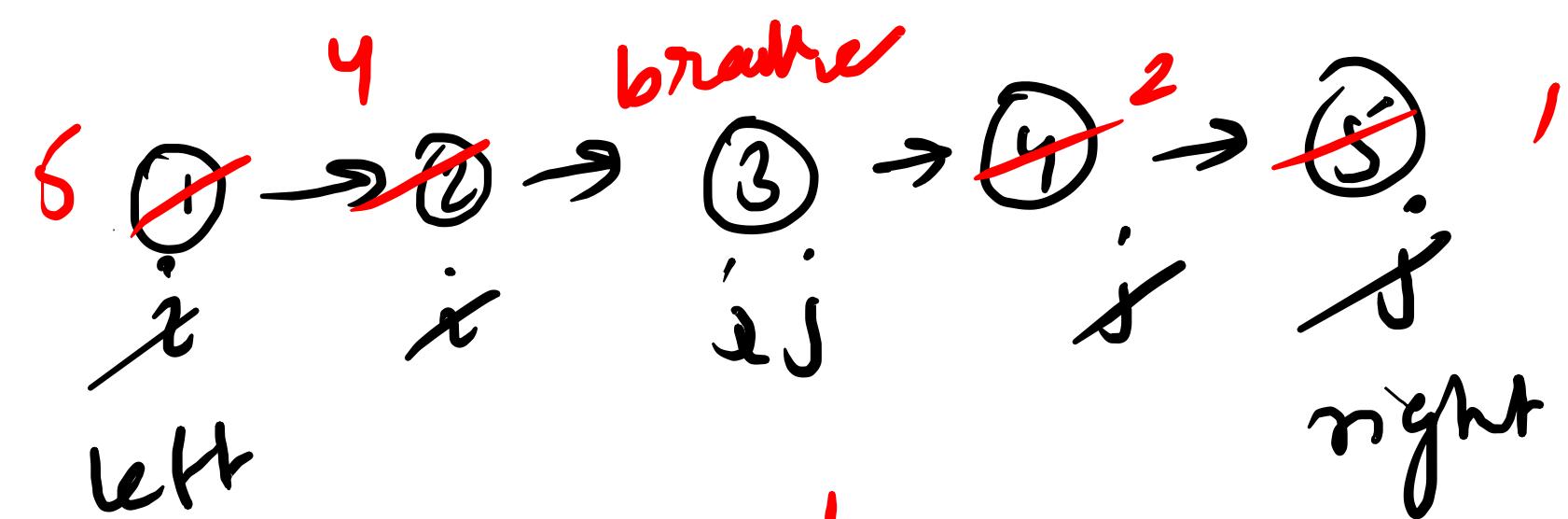
⇒

while ($i < j$)

temp = v
 $i = i + 1$
 $j = temp$

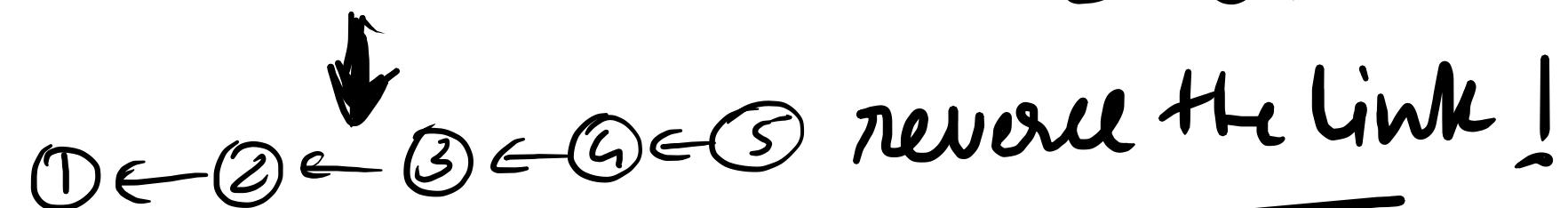
DRY RUN;
Reverse linked list (Data Iterative)

```
public void reverseDI() {  
    int i = 0;  
    int j = size - 1; Doesn't matter  
    while (i  $\leq$  j) {  
        Node left = getNodeAt(i); ✓  
        Node right = getNodeAt(j); ✓  
        int temp = left.data;  
        left.data = right.data; } swaps  
        right.data = temp;  
        i++; }  
        j--;}  
    }  
  
data swap. left ++  
right --
```

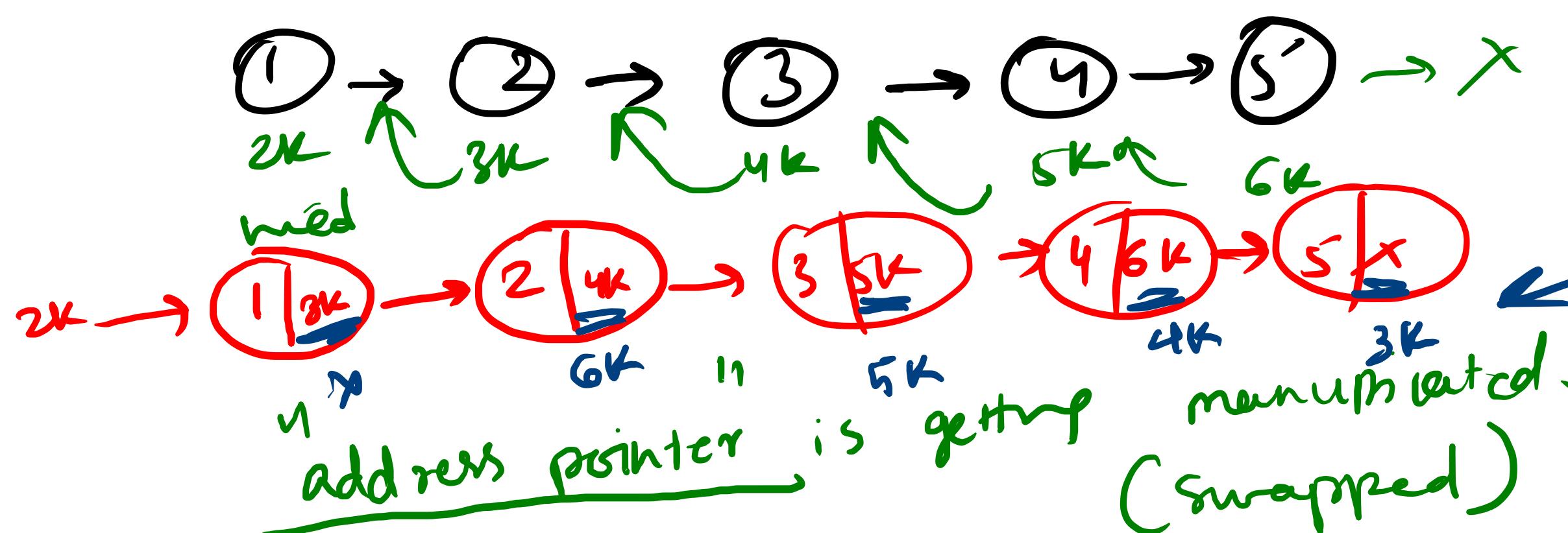


A Swap!

Problem : 2
Reverse LL(pointer iteratively)



the
catch:



NOTE:

Address
gets
swapped

Links Swapped.

Pointer iterative (LL · reversal). :-

while ($i \leq j$) :

Node temp = head.next

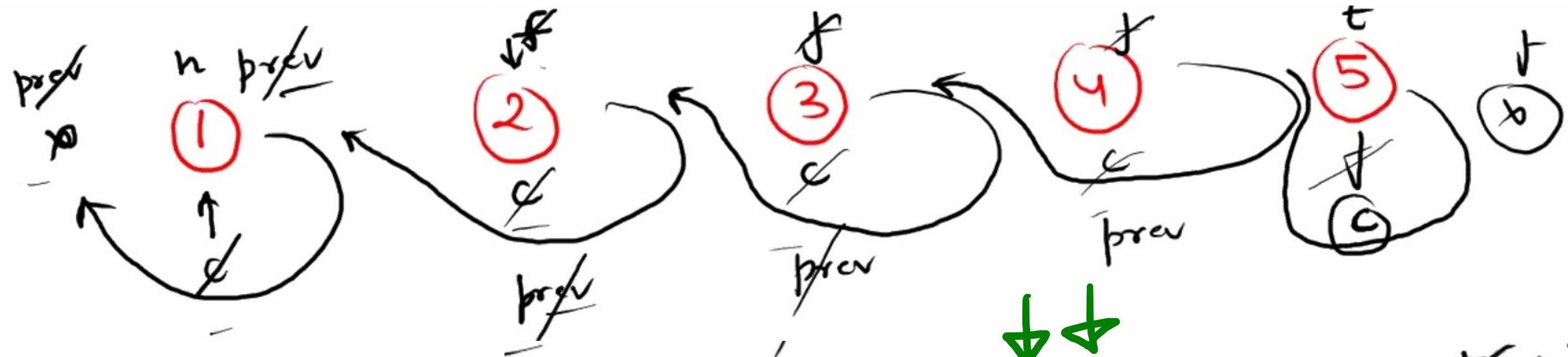
head.next = tail.next

tail.next = temp.

} something of this sort ??

somehow
head ptr ++
tail ptr --

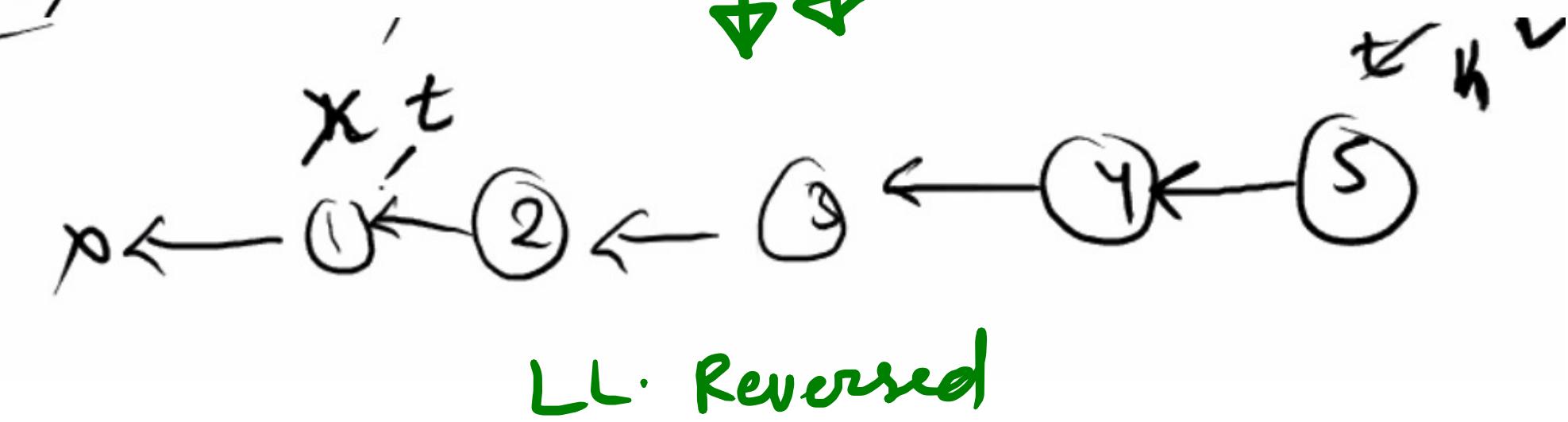
Problem 2: Approach



equations:

$$p = \text{null}$$
$$c = \text{head}$$

$$\begin{cases} t = c \rightarrow \text{next} \\ c \cdot \text{next} \leftarrow t = p; \\ p = c \\ c = t \end{cases}$$



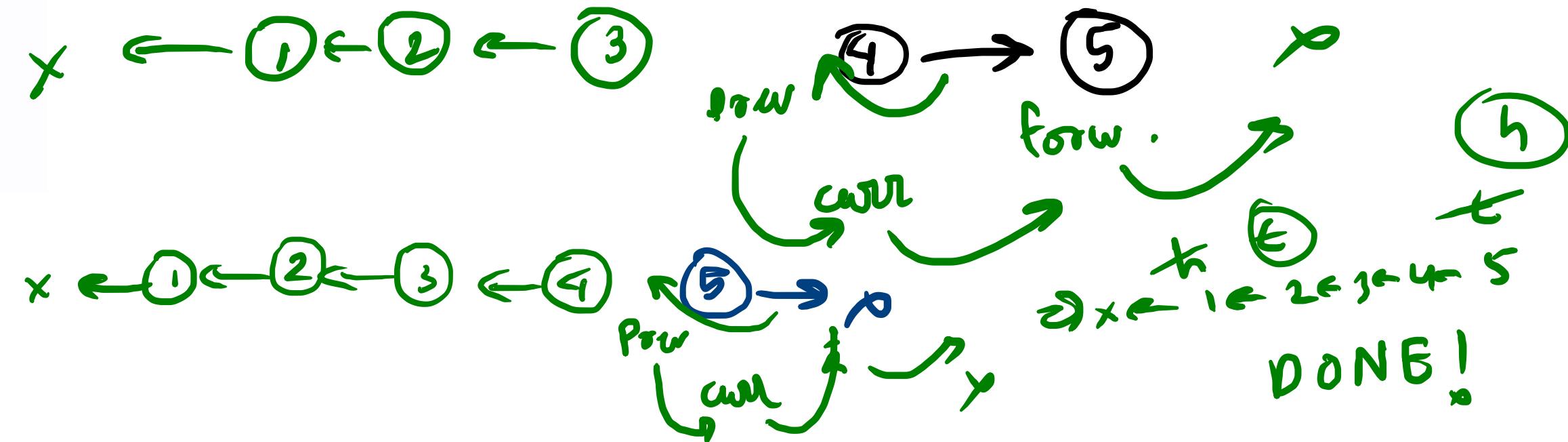
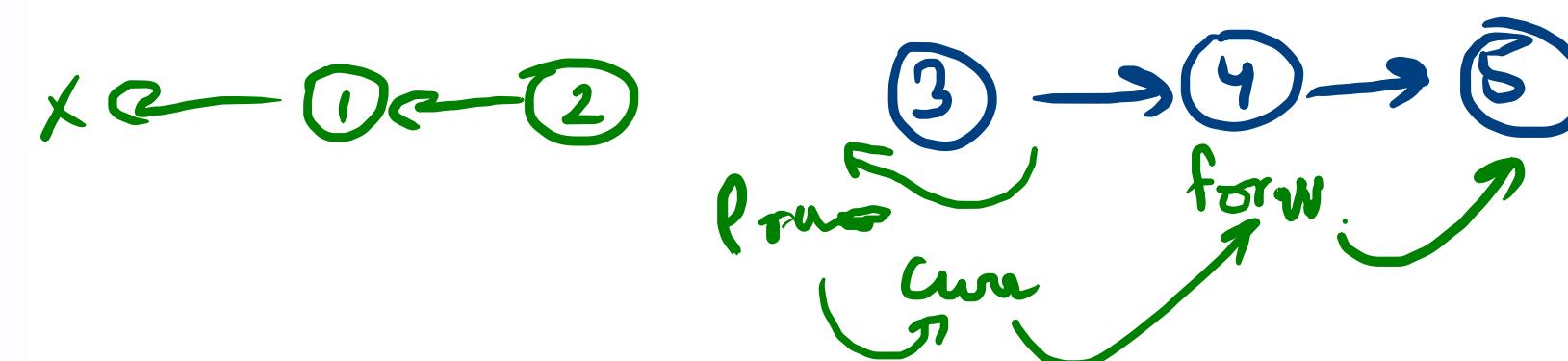
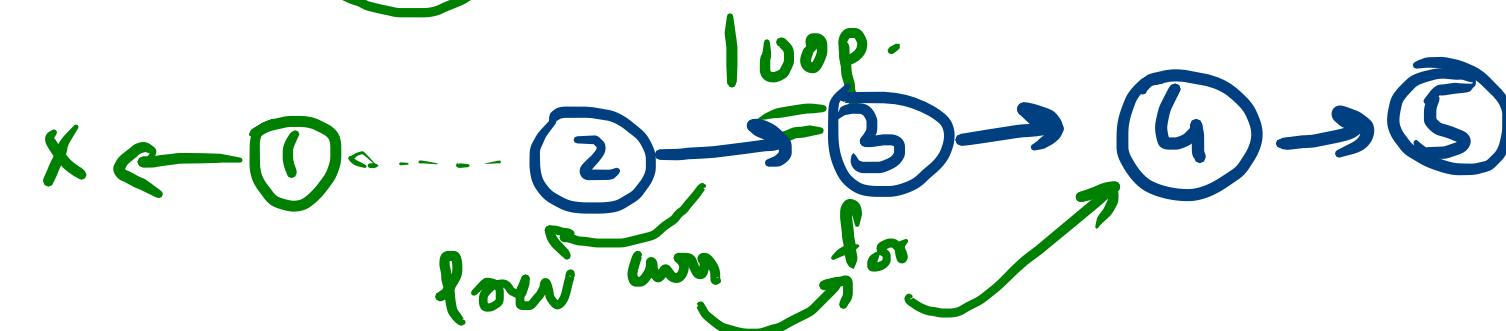
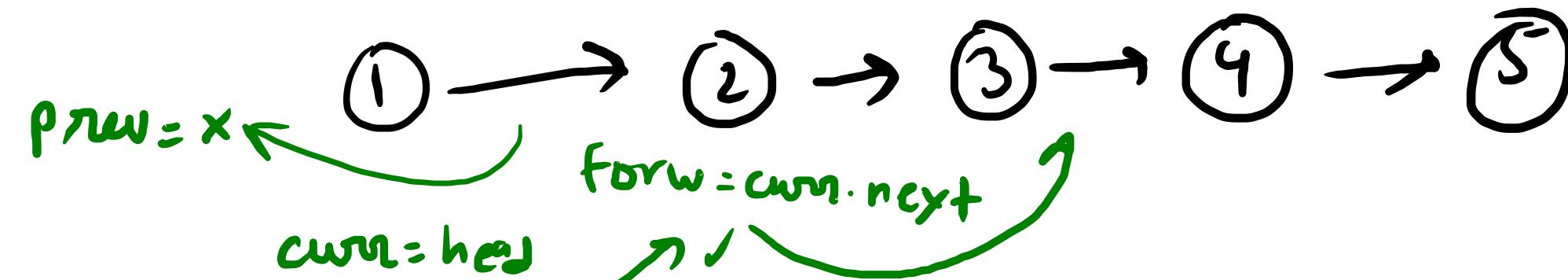
Dry Run: Problem 2

Reverse PR. iterative .

```
public void reversePI(){
    Node prev = null;
    Node curr = head;

    while(curr != null){    /|
        Node forw = curr.next;
        curr.next = prev;
        prev = curr;
        curr = forw;
    }

    Node temp = head;
    head = tail;
    tail = temp;
}
```



You are viewing Aman Srivastava's screen | View Options | Undo | Redo | Pages | Previous | Next | Erase | Board | Web | Documents | Show Desktop | OpenBoard | View

```

public void reversePI(){
    Node prev = null;
    Node curr = head;

    while(curr != null){
        Node forw = curr.next;
        curr.next = prev;
        prev = curr;
        curr = forw;
    }

    Node temp = head;
    head = tail;
    tail = temp;
}
  
```

size=5.

while (curr != null)

$te = h$
 $h = tail$
 $tail = temp\ h$

1.4 x

Unmute Start Video Participants 142 Chat Share Screen Record Reactions Apps Leave

You are viewing Aman Srivastava's screen

Recording

Stylus Color Line Eraser Backgrounds Undo Redo Pages Previous Next Erase Board Web Documents Show Desktop OpenBoard

Stack

LIFO
Last in first out

Bucket

+

Queue

FIFO
First in first out

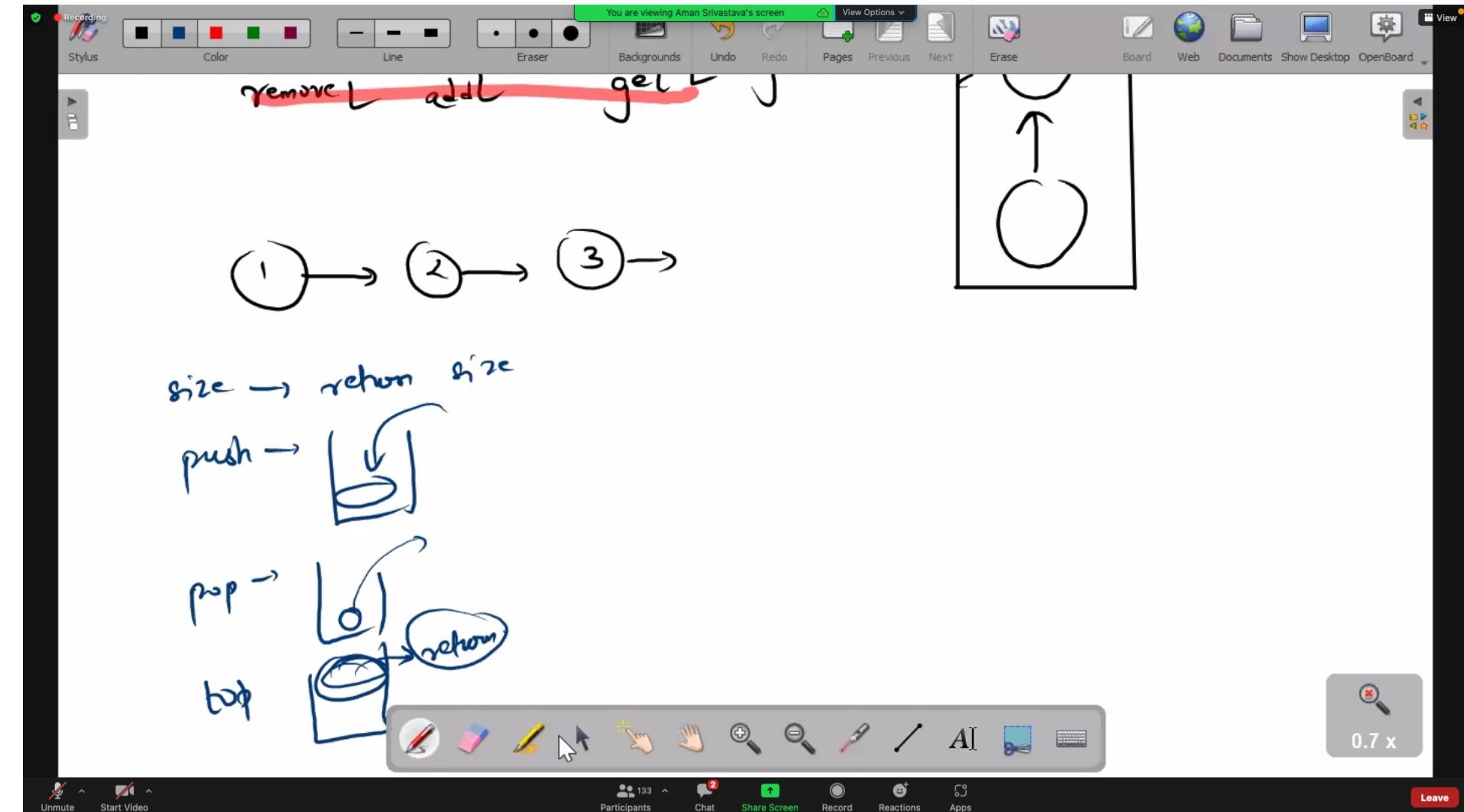
Pipe

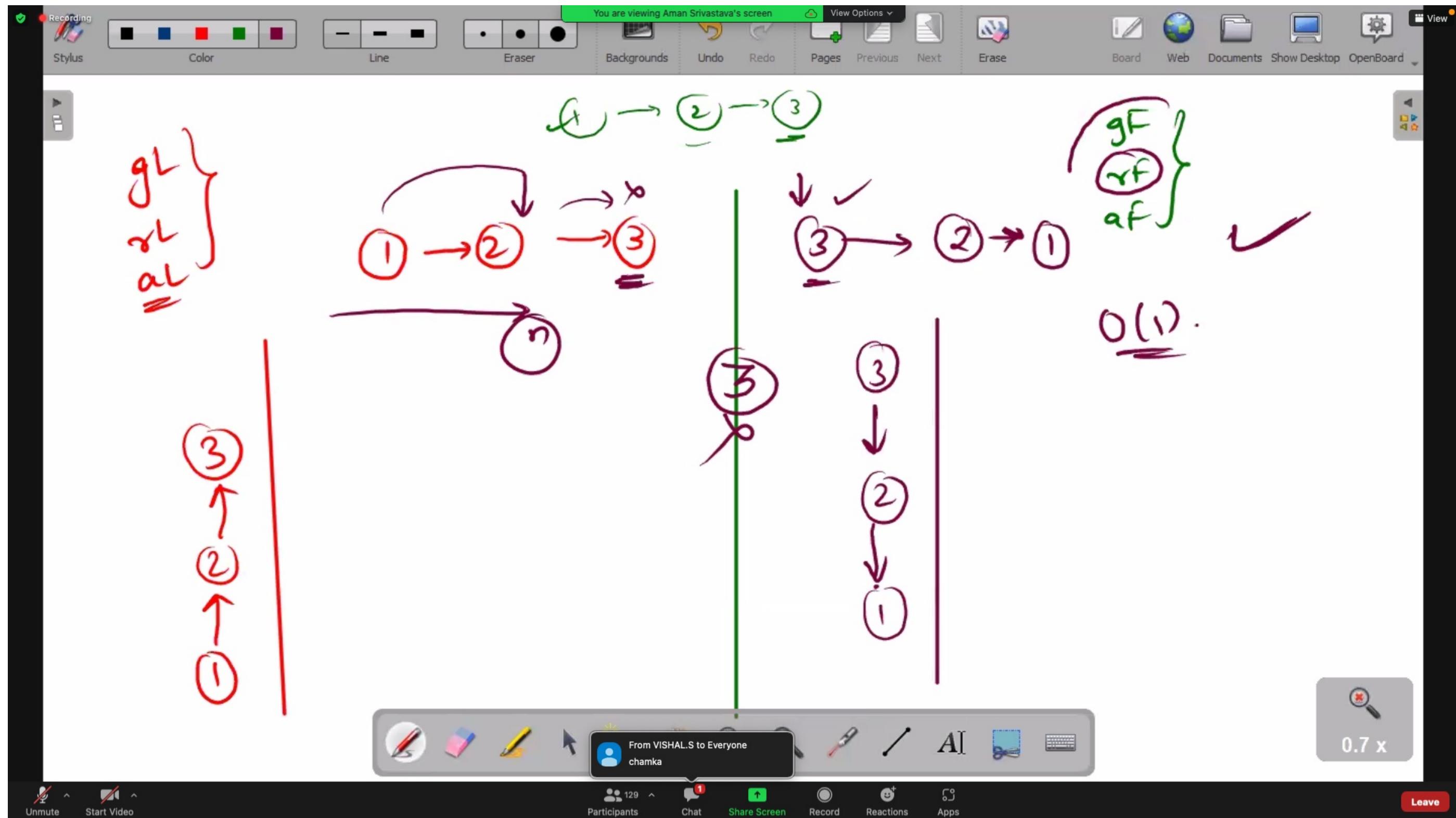
1.4 x

Unmute Start Video Participants Chat Share Screen Record Reactions Apps Leave

linked list to stack Adapter: (Problem 3)

Make Stack wrong LL.





Recording

```
1  PUBLIC static class LLToStackAdapter {  
2      LinkedList<Integer> list;  
3  
4  
5      public LLToStackAdapter() {  
6          list = new LinkedList<>();  
7      }  
8  
9  
10     int size() {  
11         return list.size();  
12     }  
13  
14     void push(int val) {  
15         list.addFirst(val);  
16     }  
17  
18     int pop() {  
19         return list.removeFirst();  
20     }  
21  
22     int top() {  
23         return list.getFirst();  
24     }  
25  
26     public static void main(String[] args) throws Exception {  
27         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  
28         LLToStackAdapter st = new LLToStackAdapter();  
29  
30         String str = br.readLine();  
31         String[] arr = str.split(" ");  
32         for (String s : arr) {  
33             st.push(Integer.parseInt(s));  
34         }  
35         System.out.println(st.pop());  
36         System.out.println(st.top());  
37     }  
38 }
```

You are viewing Aman Srivastava's screen

View Options

View

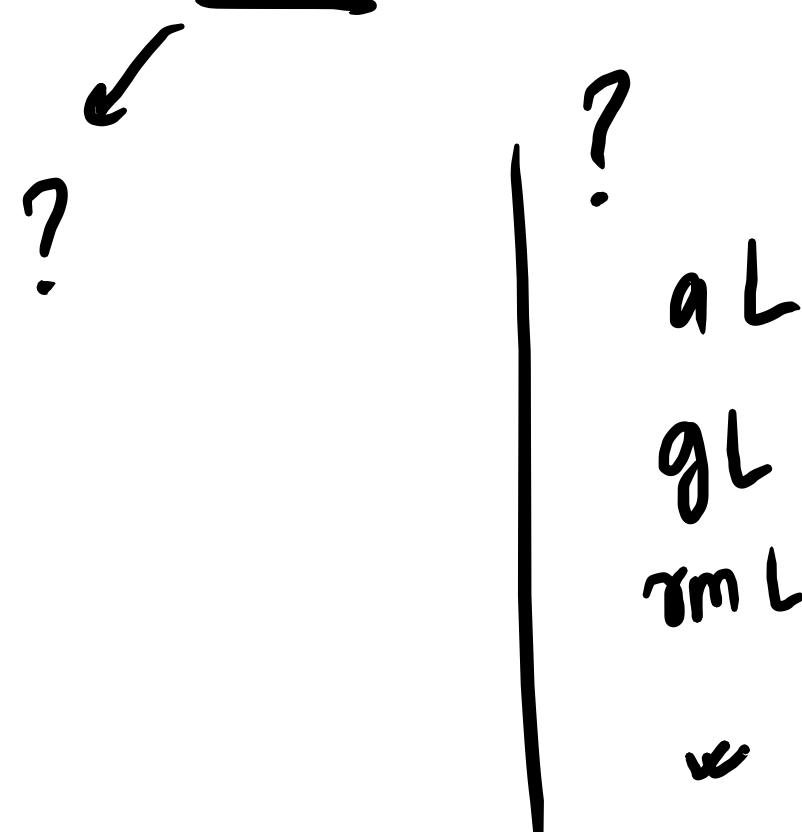
Participants: 123 Chat: 1 Share Screen Record Reactions Apps

Unmute Start Video

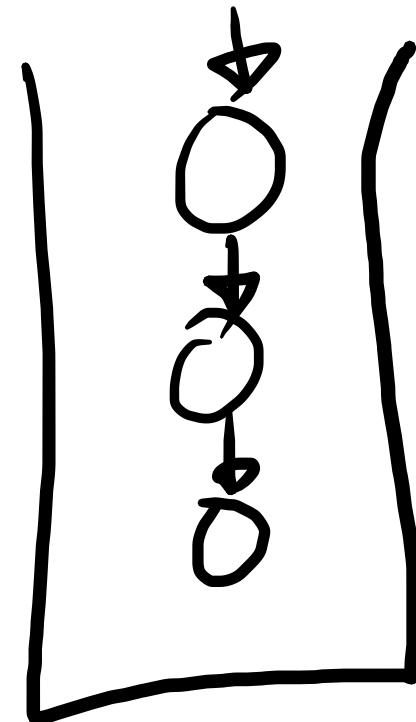
linked list to Queue Adapter: ~~CO-DL~~

FIFO

aF
gF
mF



size
add
remove
peek



Hint!
Attempt
@
H/W.

k^{th} Node from Last

Approach

$k=3 \rightarrow \text{Node}$

* Slow + fast pointer (gap strategy)

From Scylla23 to Everyone
y

Participants: 102 Chat: 1 Share Screen Record Reactions Apps

Unmute Start Video

Leave

0.5 x

