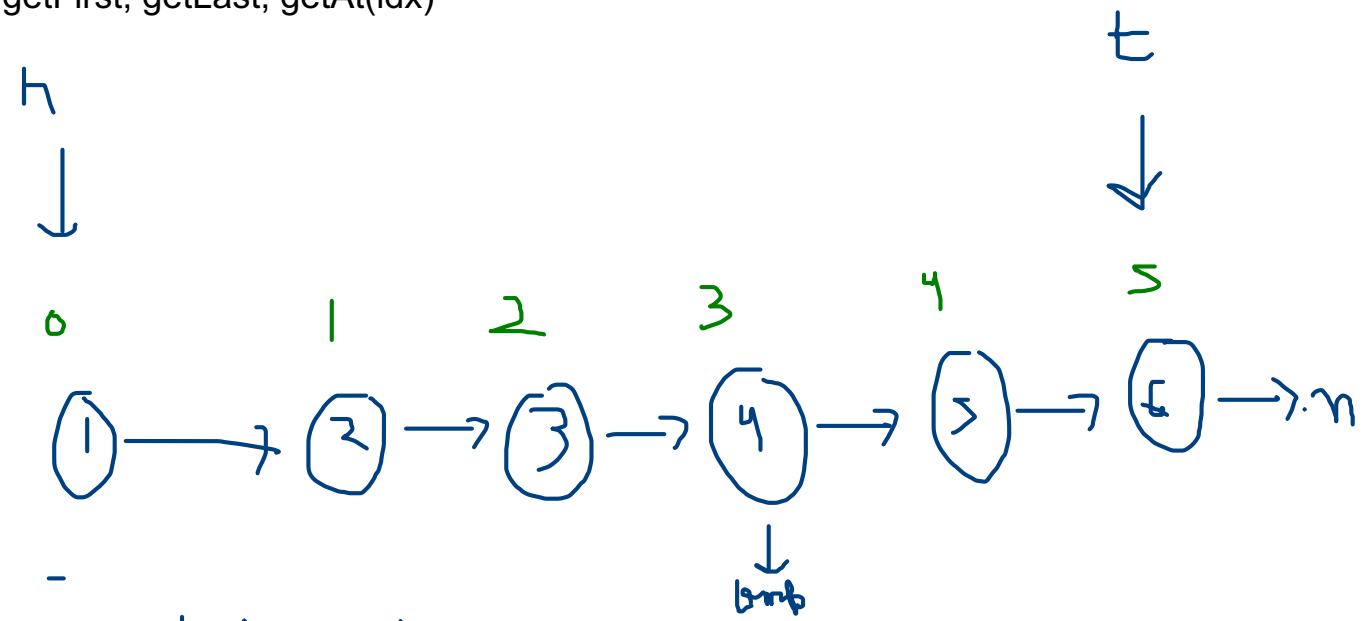
getFirst, getLast, getAt(idx)



getFirst ? getLast ?

getAt(3)

idx = ~~3~~ 2 + 0

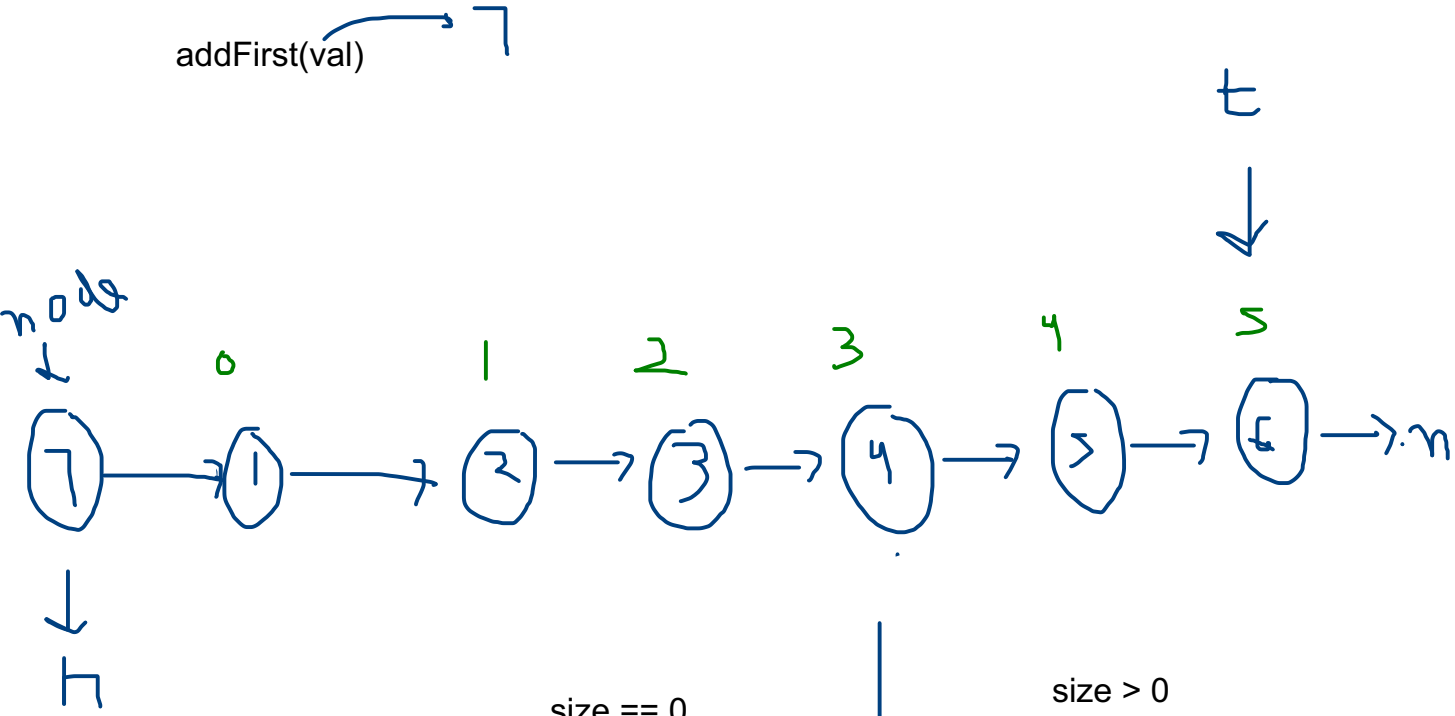
invalid case

idx = 8

size = 6

0 to 5

addFirst(val)



size == 0

make node

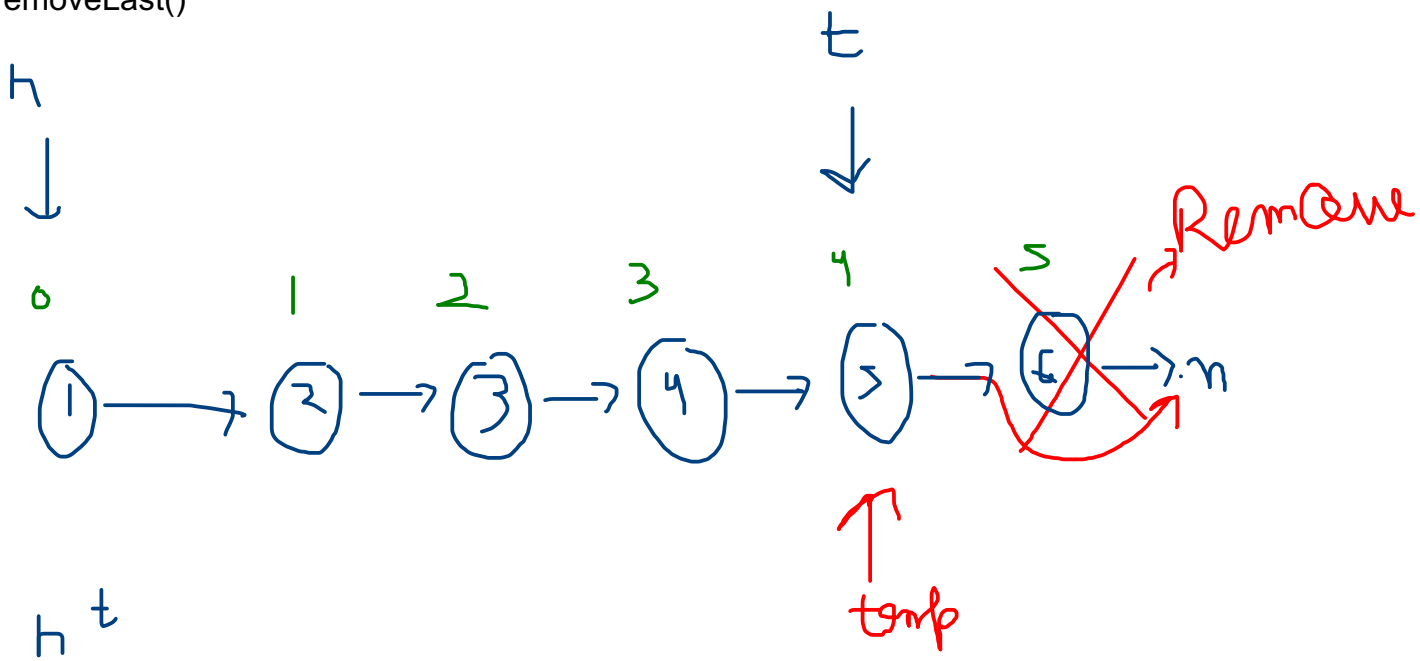
head = tail = node

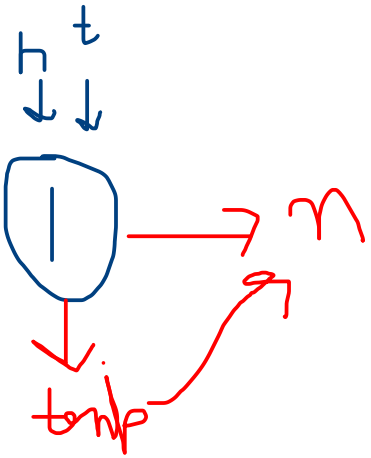
size > 0

make node

node.next = head
head = node

removeLast()

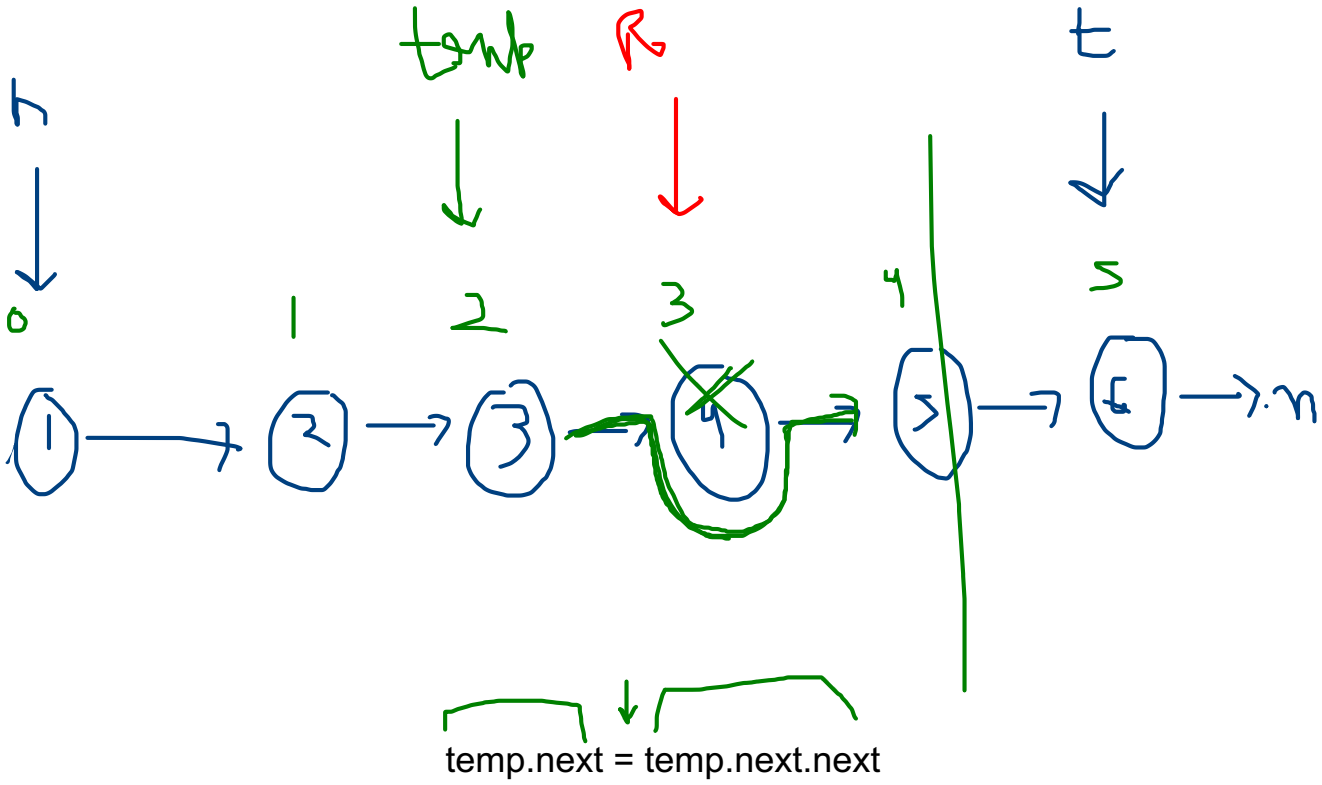




case 1	case 2	case 3
size == 0	size == 1	size > 1
print list empty	head = tail = null	temp.next.next == null --> 2nd last element temp.next = null

removeAt(idx)

idx = 3



Different Types of LinkedList

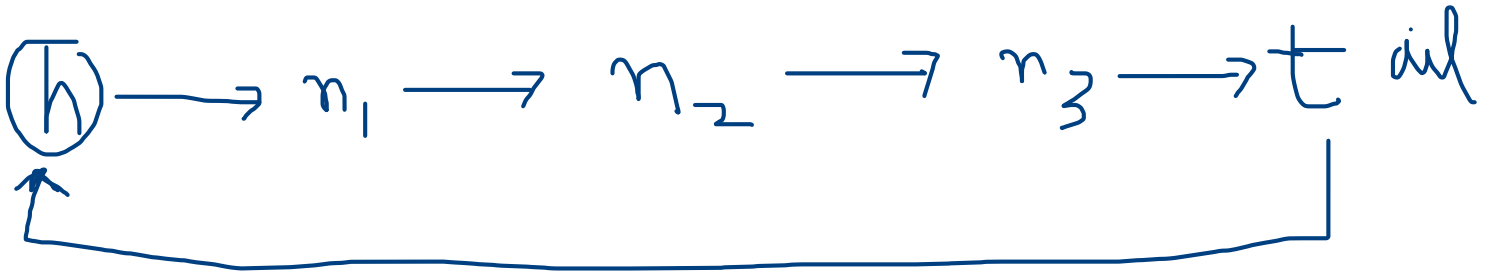
1. Singly LinkedList

head --> n1 --> n2 --> n3 --> tail --> null

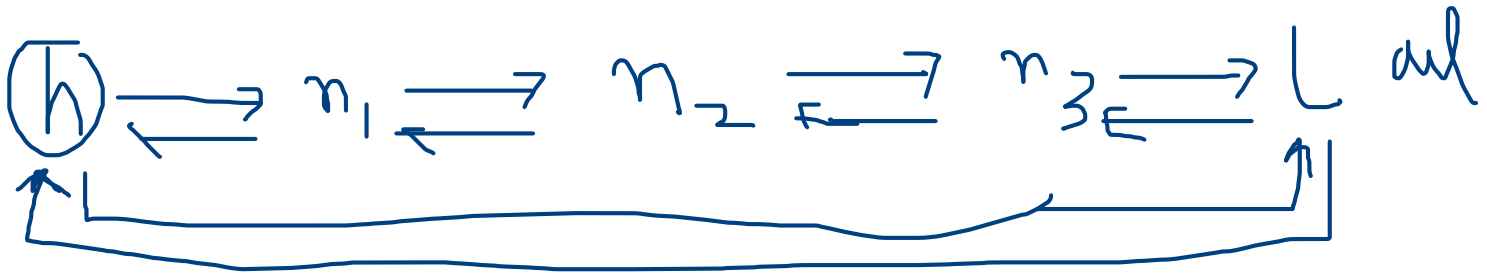
2. Doubly LinkedList

head --> n1 --> n2 --> n3 --> tail --> null

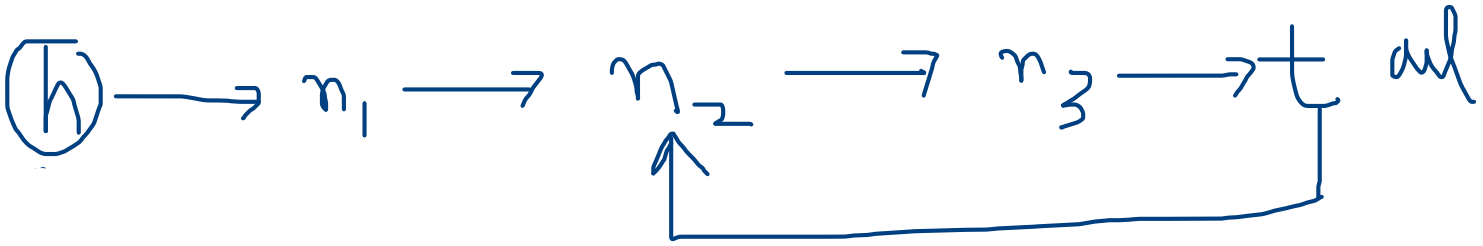
3. Circular LinkedList



4. Circular Doubly LL



5. Cyclic LL

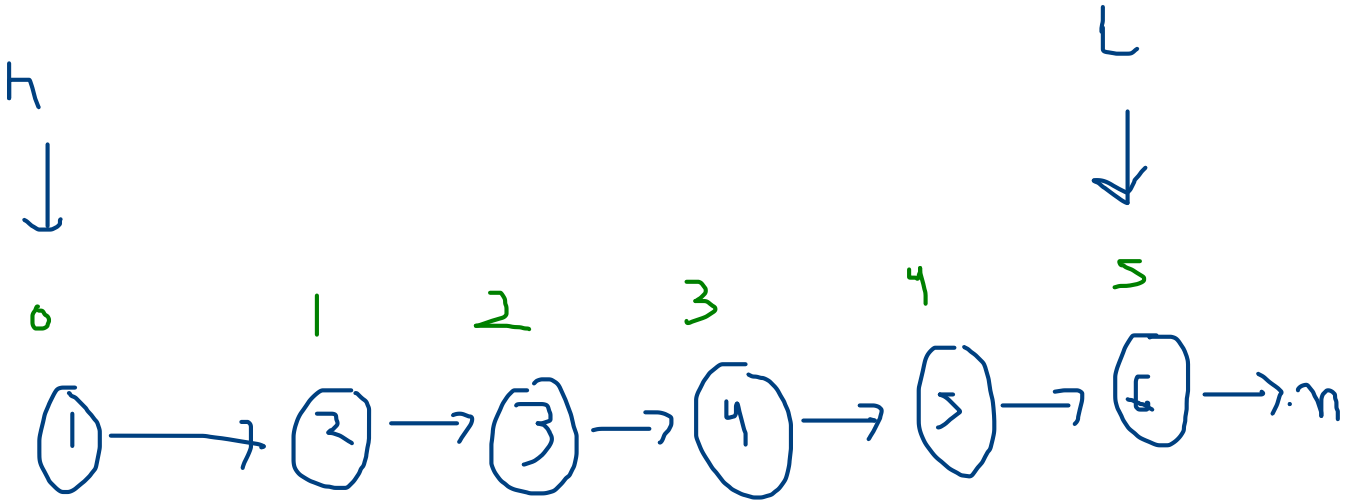


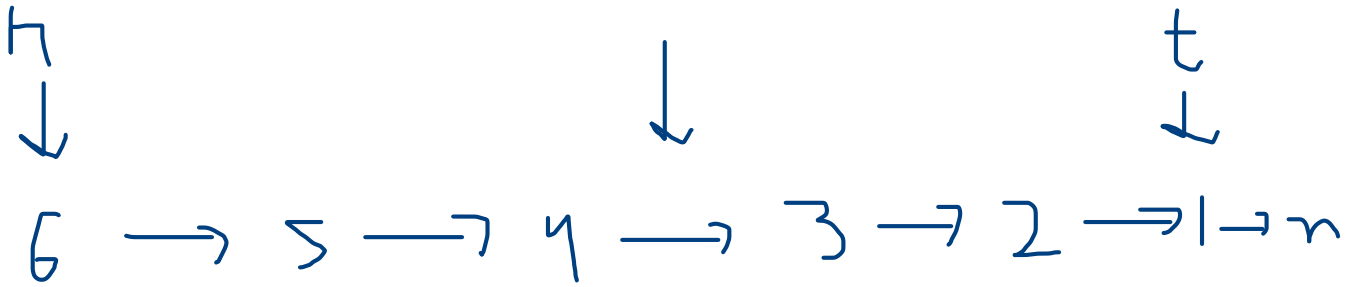
Example of Doubly LL

Mobile Phones --> Recent apps

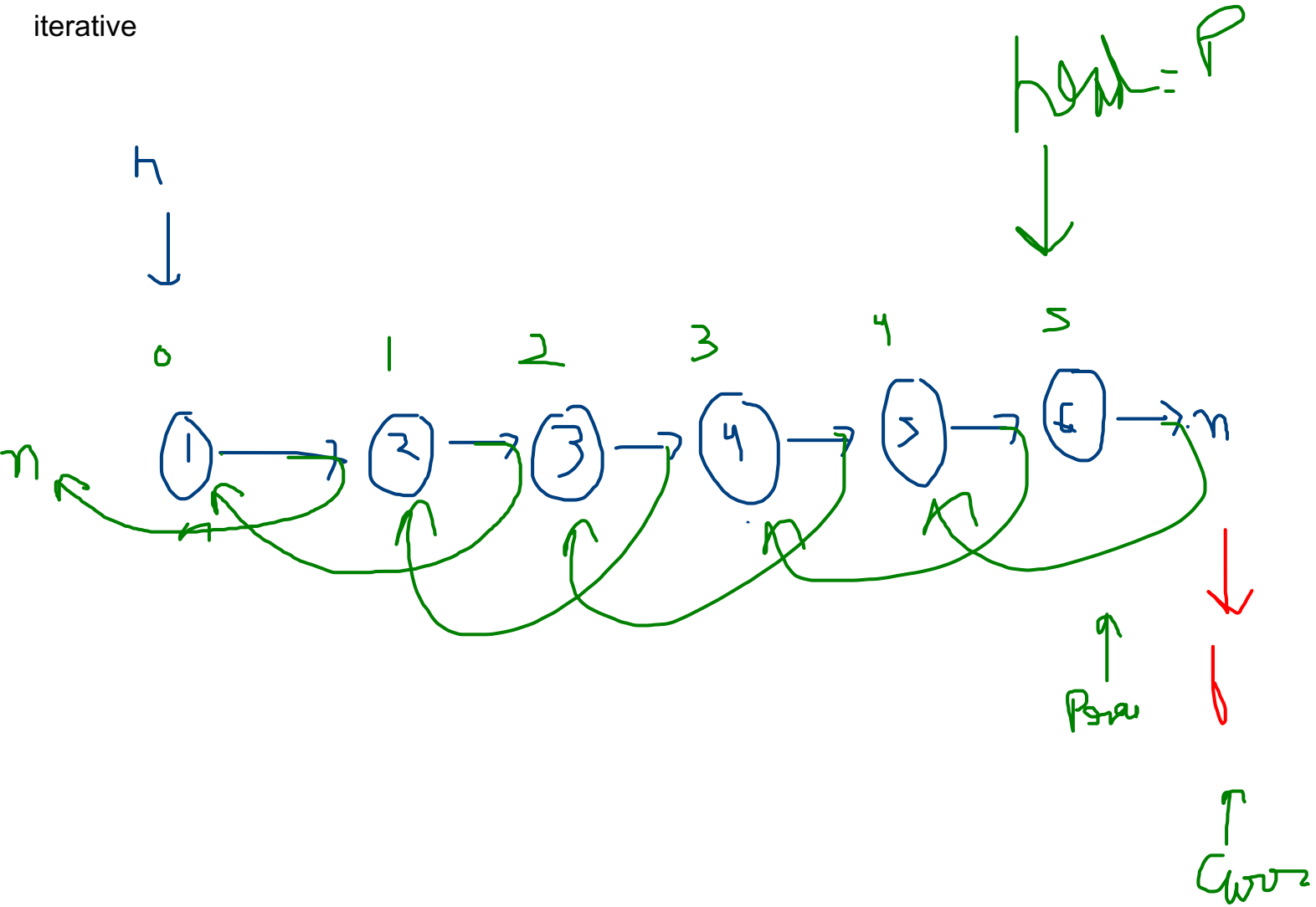
LRU Cache

Reverse LL



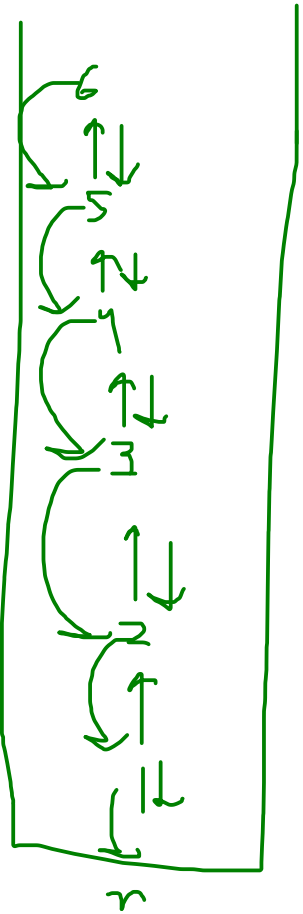


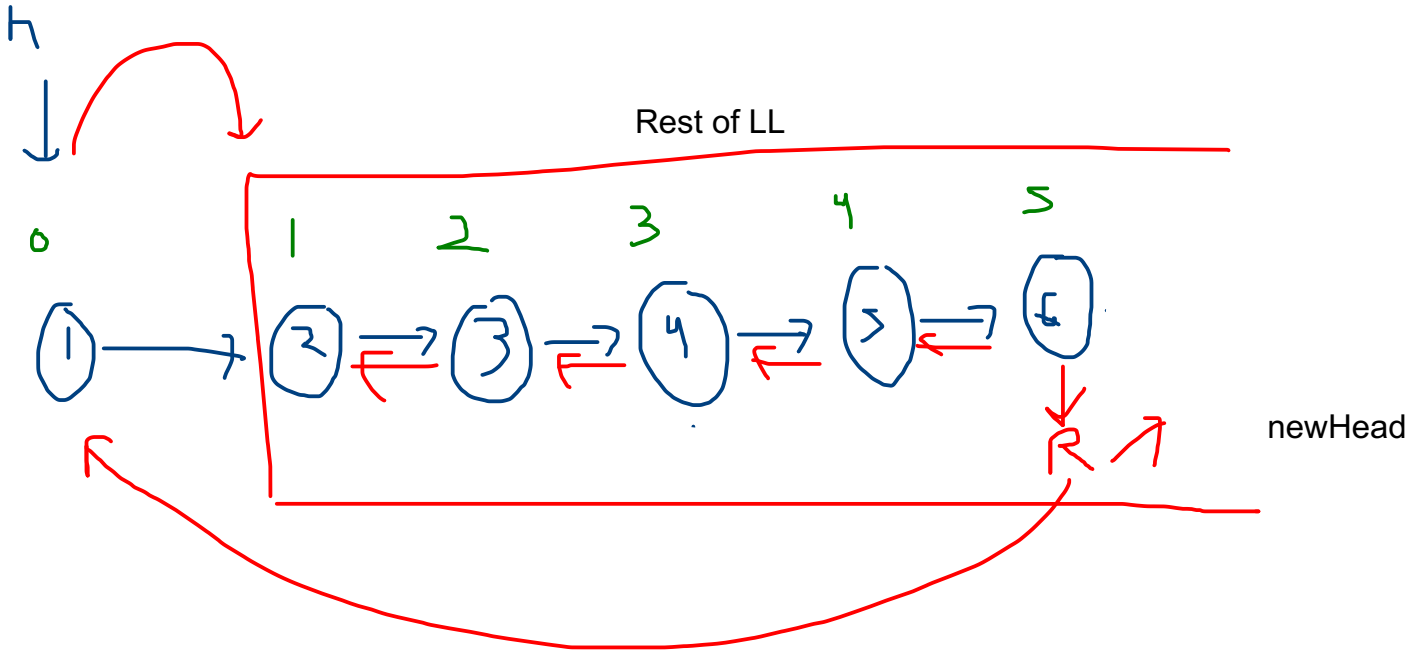
iterative



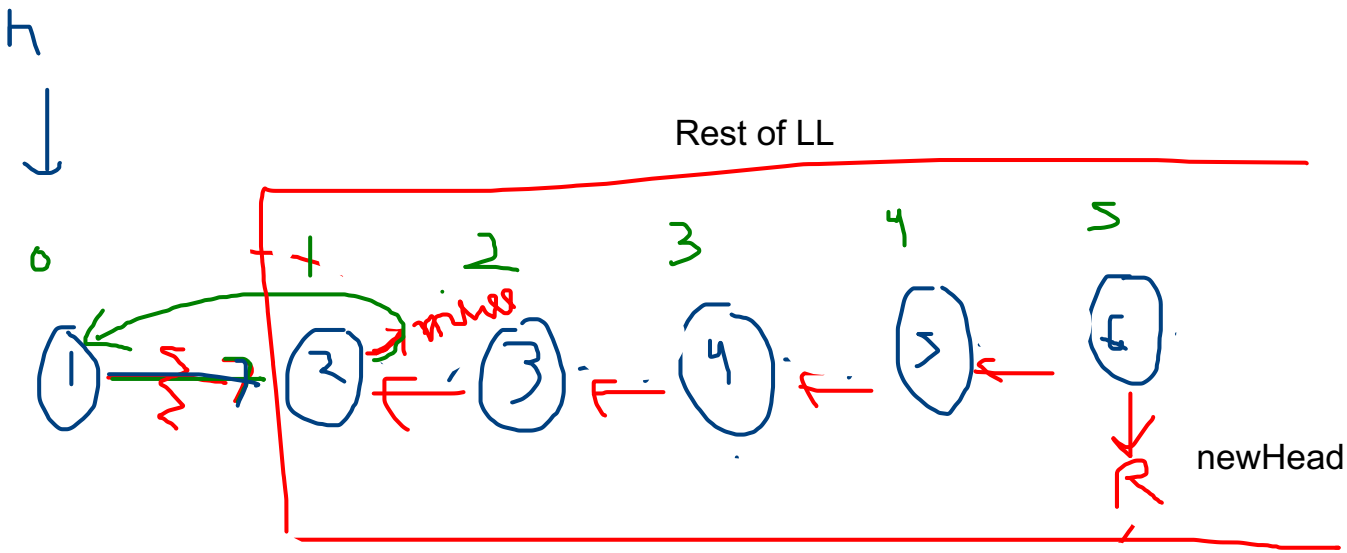
Recursion Method

Step 1





Step2



head == null || head.next == null

return head

head.next.next = head

head.next = null

return R