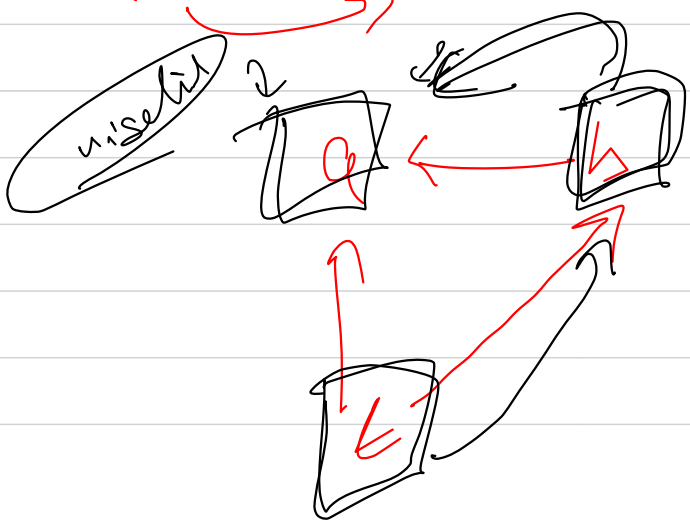
Parent

$a \longleftarrow b \longleftarrow c \longleftarrow d$

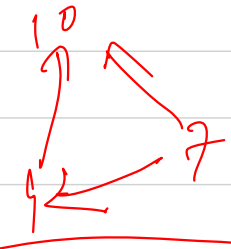
$a \longleftarrow b$



integer

$a < 4$

$a > 5$

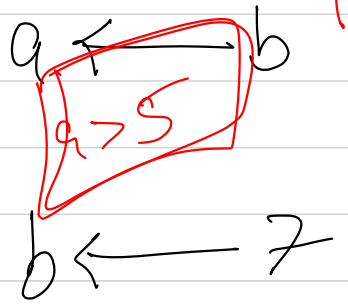


valid

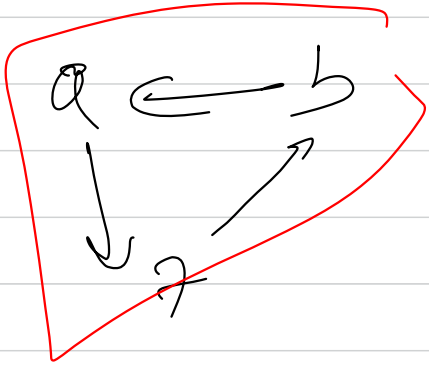
no cycle invalid

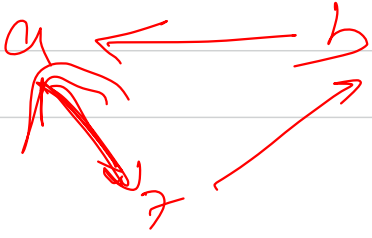
$a < 4$

$a >$



$a > 5$





$a = 2$

$a > 7$

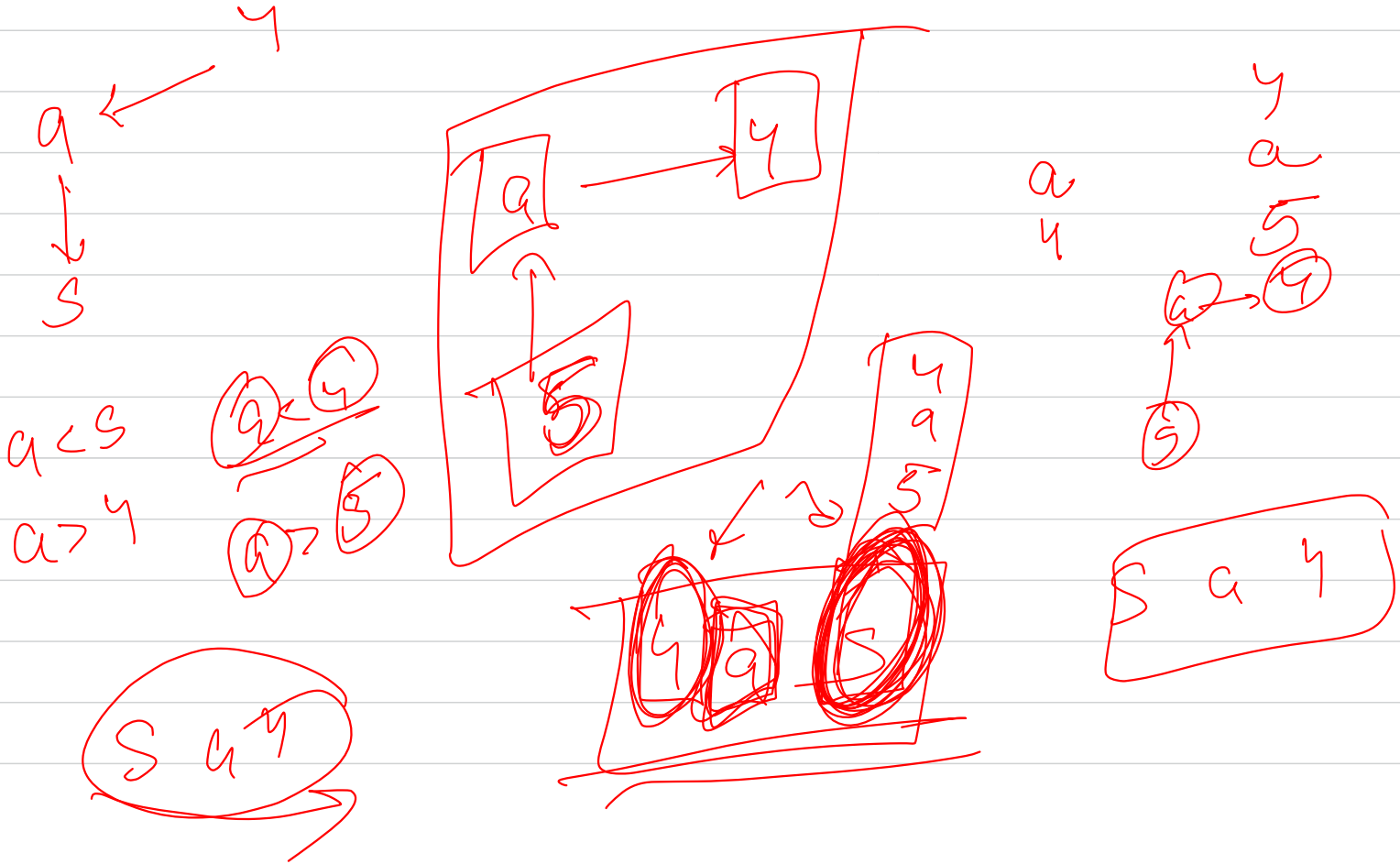
$b > 7$

$a > b$

$9 > 7$

$10 > 7$

$10 > 9$


a<s
a>y
s a y

$4 > 5$

$a < 4$

$a \to 5$

4

$a > 5$

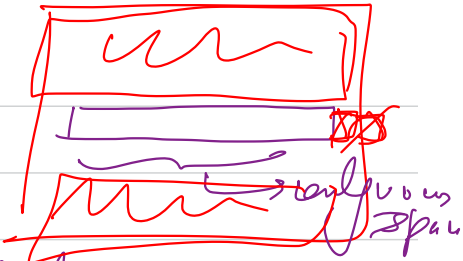
$a < 4$

$a \to 5$

4

$5 > 4$

Linked Lists



Why do we even need a new DS called as LL.

Arrays

- we need to give size of arrays
- arrays always have contiguous memory alloc

list/vectors/dorcyled

→ they also consume contiguous memory slots

Arrays doesn't consume the available memory efficiently

→ To optimize this we have got linked lists

What is a LL??

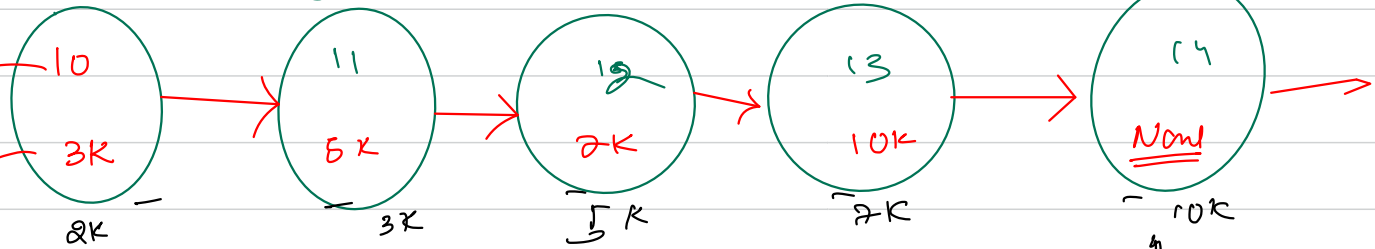
→ LL are linear data structure that store one data element in an entity called as a Node and multiple such nodes are connected to form a chained list. The nodes are not reqd to be created in continuous memory blocks, so LL never consumes a contiguous memory block.

↳ singly linked list

node → class

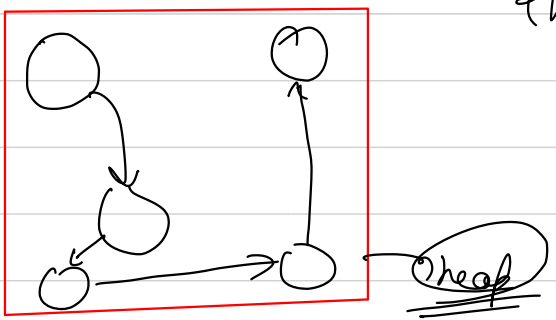
data

address of next node



The first node of LL is called as "Head"

The last node of LL is called as "Tail"



↳ memory view

The button of prev and next page in browser are implemented using LL.

- Music player software

→ cache mechanism

→ HashMap

⋮

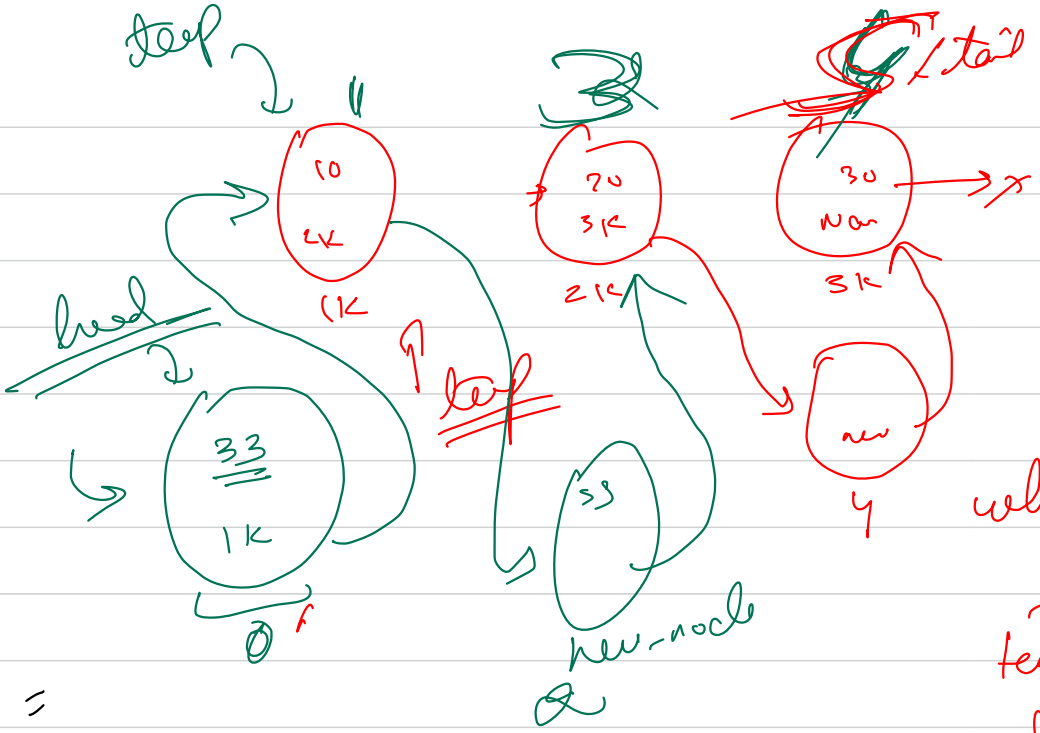
↳

```
class Node :
    data    # value of node
    next    # address of next node
```

Adding data in LL

- → at the head → $O(1)$
- → at the tail → $O(n)$
- → in middle somewhere → $O(n)$

list →
- at at 0th nd → $O(n)$
- add at last ele → $\Theta(1)$
- middle → $O(n)$



while (temp != Nan)

temp.data

temp = temp.next

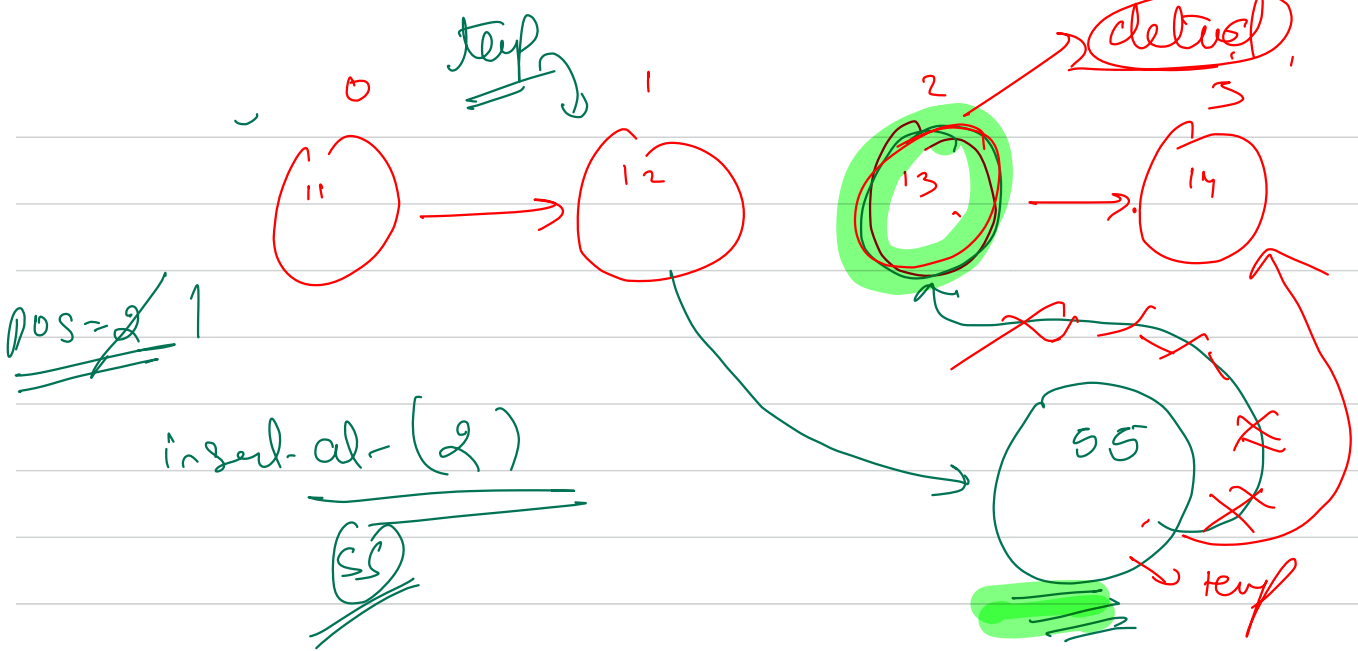
new_node.next = temp.next

temp.next = new_node

→ fetch the node previous to the idx where we want to insert

→ mark the next of new node = to the node present at the idx currently (temp.next)

→ update temp.next = new_node.



temp.next = temp.next.next

temp.next =

delete from head

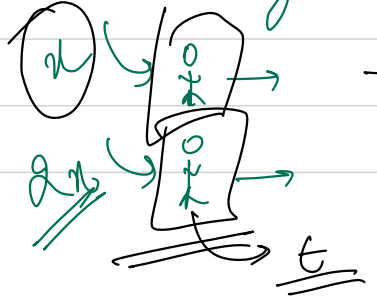
delete from tail

delete in between

Q: find mid node of a LL without using length function & by iterating only once on the LL.

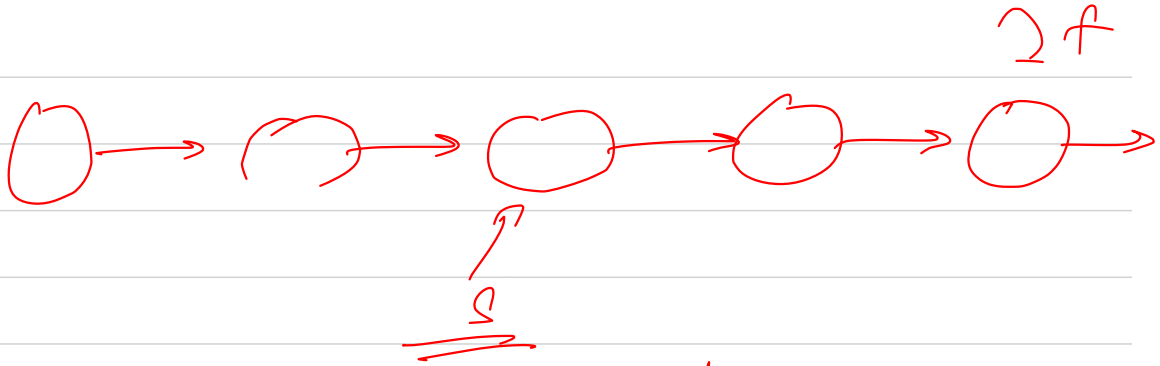
TC → $O(n)$ SC → $O(1)$

→ length of LL → L



→ $d = s \times t$ → $x \times \frac{L}{2x}$ → $\boxed{\frac{L}{2}}$

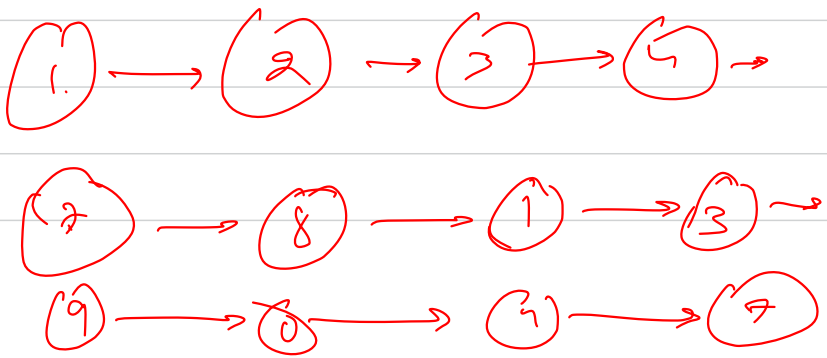
$$\boxed{t = \frac{L}{2x}}$$



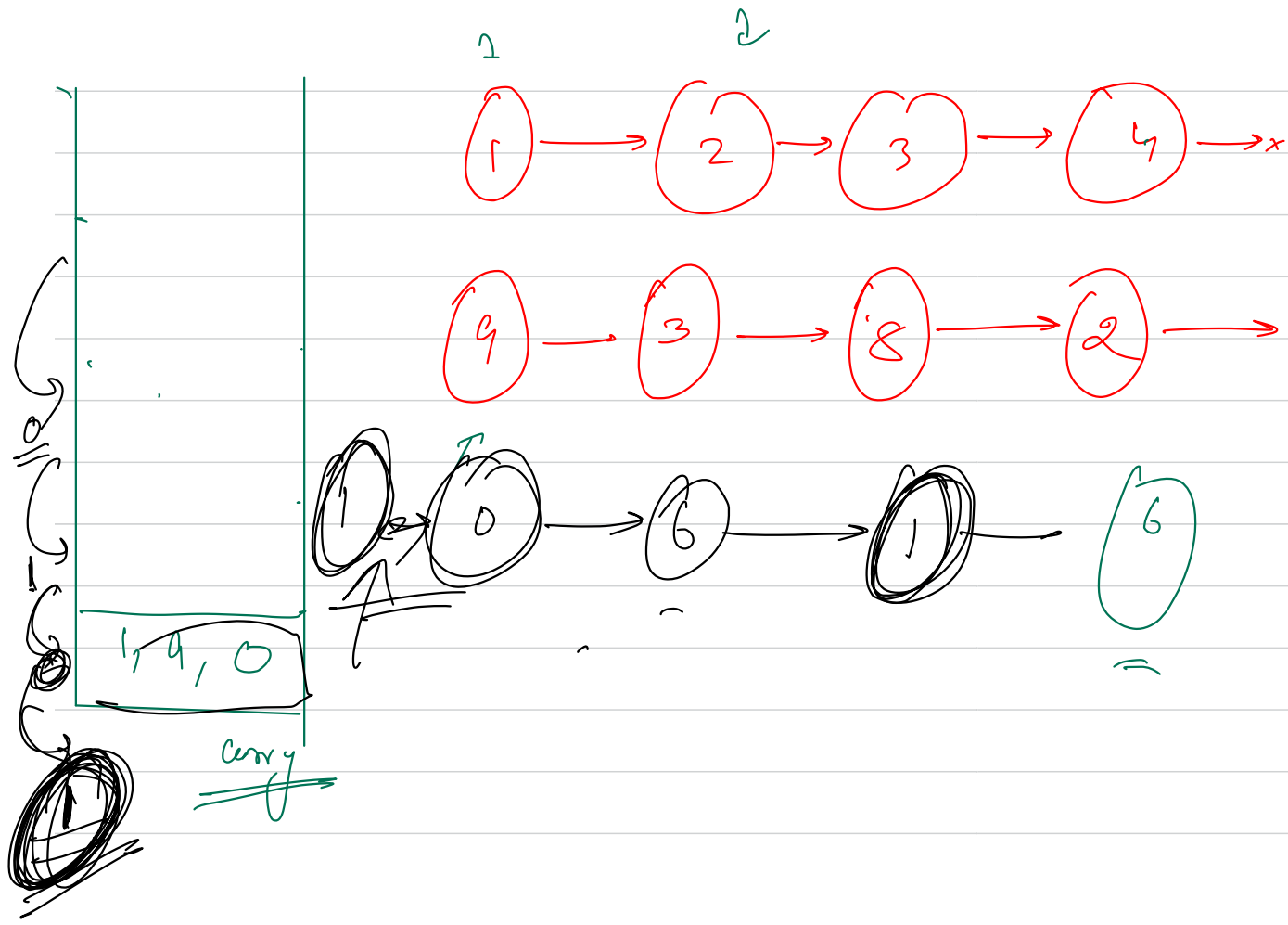
Q: You're given 2 linked lists of digit, of same size.

Add them & return a new ll.

single pass



1
1 2 3 4
2 8 1 3
9 0 4 7



Recursion

get me prev state from which my caller