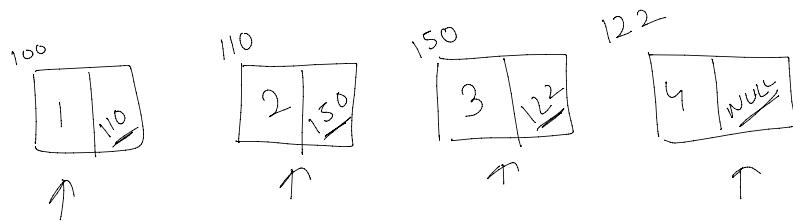
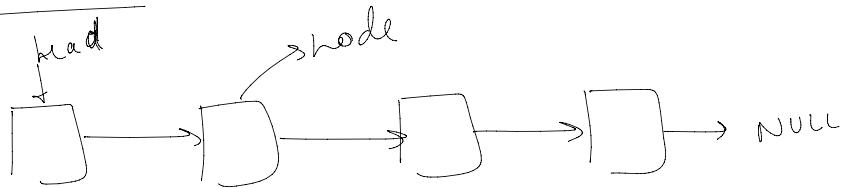
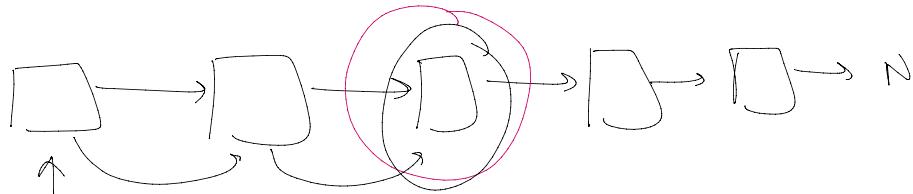


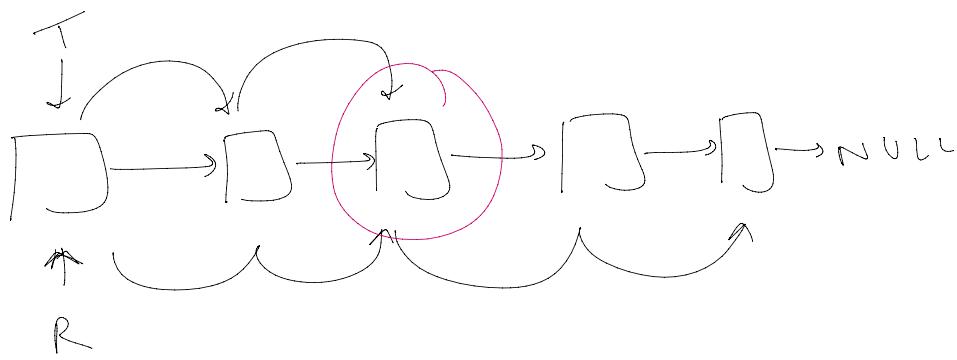
Linked lists100  
↑

1, 2, 3, 4

access an index (array) :  $O(1)$ access a node (linked list) :  $O(n)$ Middle of a linked list

→ traverse the linked list and count the number of nodes. → I traversed

- 1) Traverse the linked list and count the number of nodes.  $\rightarrow$  1 traversal
- 2) Traverse half the length.  $\rightarrow$  1 traversal



- 1) Initialize two pointers, Rabbit & tortoise to head.
- 2) Rabbit moves 2 places while tortoise moves one.
- 3) When Rabbit reaches the end, tortoise will be at the middle.
- 4) Return the tortoise.

$r = \text{head};$

$t = \text{head};$

$\text{while}(r \neq \text{NULL} \text{ and } r \rightarrow \text{next} \neq \text{NULL}) \{$

$r = r \rightarrow \text{next} \rightarrow \text{next};$

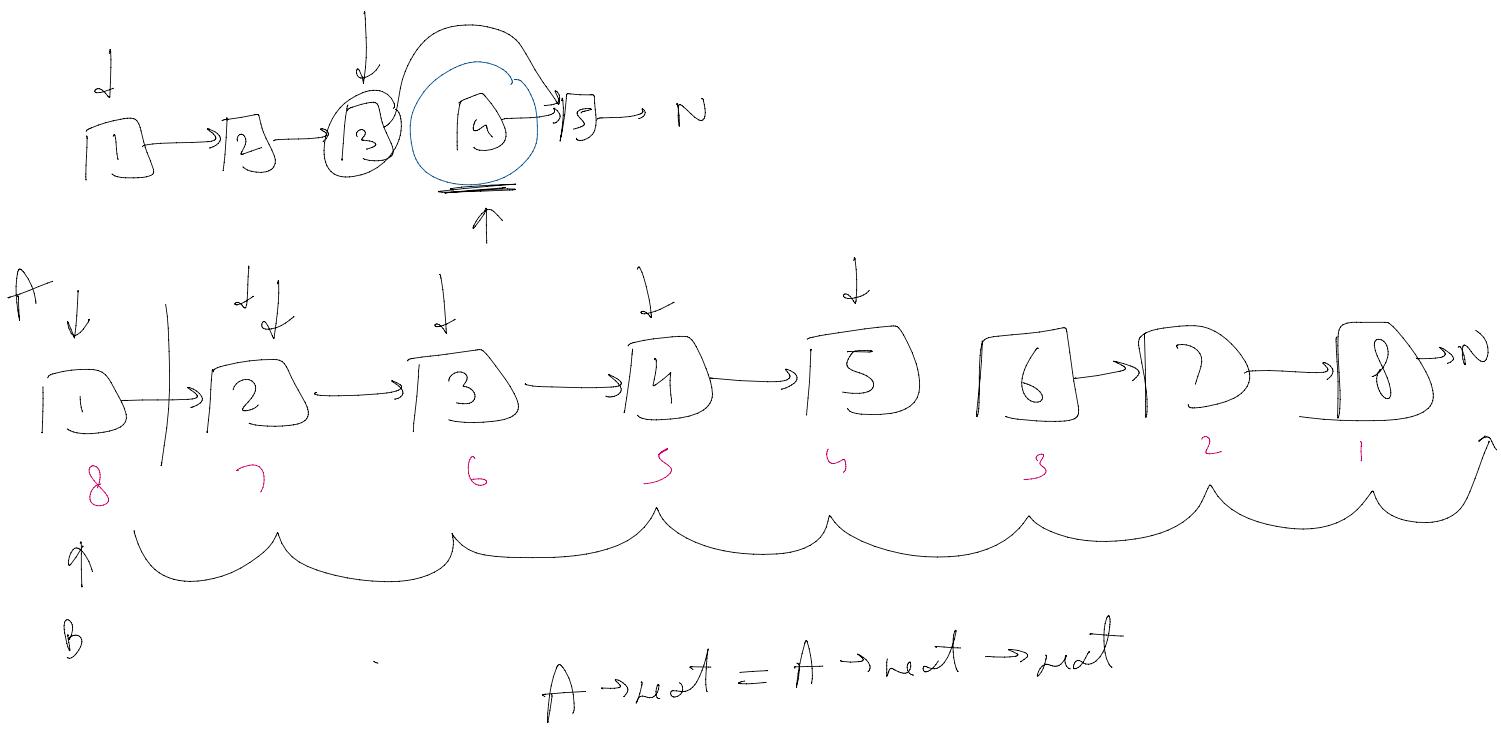
$t = t \rightarrow \text{next};$

3

return tortoise

Remove  $n^{\text{th}}$  node from end of a linked list

How to remove an element from a linked list.



1)  $A = \text{Lead}; B = \text{Lead}$

2)  $B$  moves  $\rightarrow$  places forward.

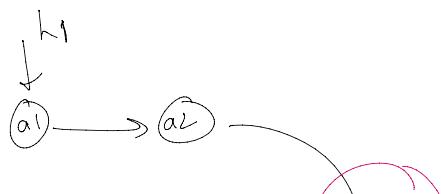
3)  $A$  &  $B$  move forward together until  $B$  reaches the end.

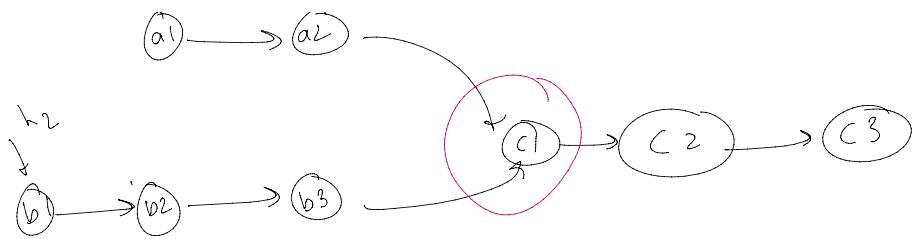
4) If  $A == \text{Lead}$ , return  $\text{Lead} \rightarrow \text{next}$ .

5)  $A \rightarrow \text{next} = A \rightarrow \text{next} \rightarrow \text{next}$ .

6) Return  $\text{Lead}$ .

Intersection of two linked lists

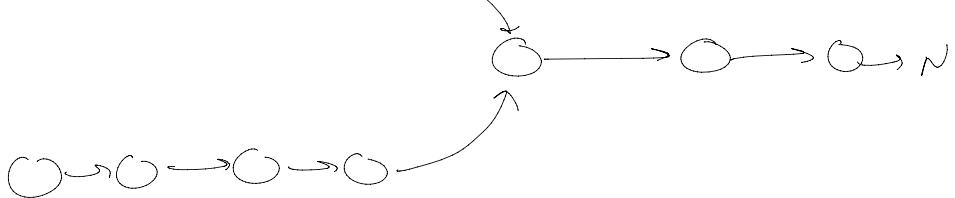




$$l_1 = 5$$

$$l_2 = 7$$

$$7 - 5 \neq 2$$



1) Find the length of the first linked list ( $l_1$ ).

2) Find the length of the second linked list ( $l_2$ ).

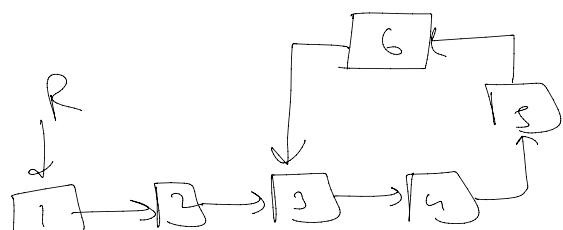
3) Move  $|l_1 - l_2|$  space forward in the longer linked list.

4) Move both the pointers forward until  $A == B$  or

$A == N$  or  $B == N \rightarrow \text{break}$ .

5) return A.

Detect loop in the linked list

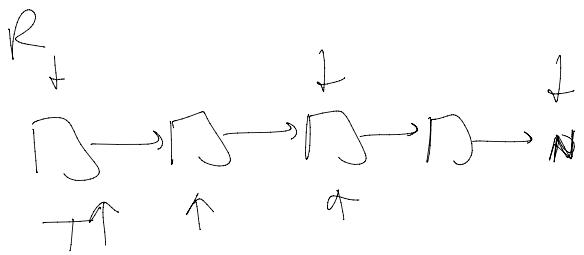
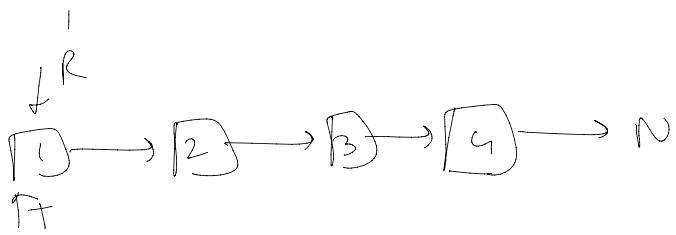


$\begin{matrix} \uparrow \\ T \\ \downarrow \\ L \end{matrix}$

$R_1$

1

1



1)  $R = \text{head}$ ,  $T = \text{head}$

2) if ( $R \rightarrow \text{next} == \text{NULL}$  ||  $R \rightarrow \text{next} \rightarrow \text{next} == \text{NULL}$ ) {  
    return false;

    3

3) do {

$R = R \rightarrow \text{next} \rightarrow \text{next};$

$T = T \rightarrow \text{next};$

    if ( $R == T$ ) return true

    3 while ( $R != \text{NULL}$  and  $R \rightarrow \text{next} != \text{NULL}$ );

        return false;