



Today's agenda

↳ Point linkedlist

↳ insert in a linkedlist

↳ delete in a linkedlist

↳ Reverse the linkedlist

↳ Find mid

↳ floyd cycle.



AlgoPrep

Linked Class

```
Class Node {
```

```
    int val;
```

```
    Node next;
```

```
Node (int v) {
```

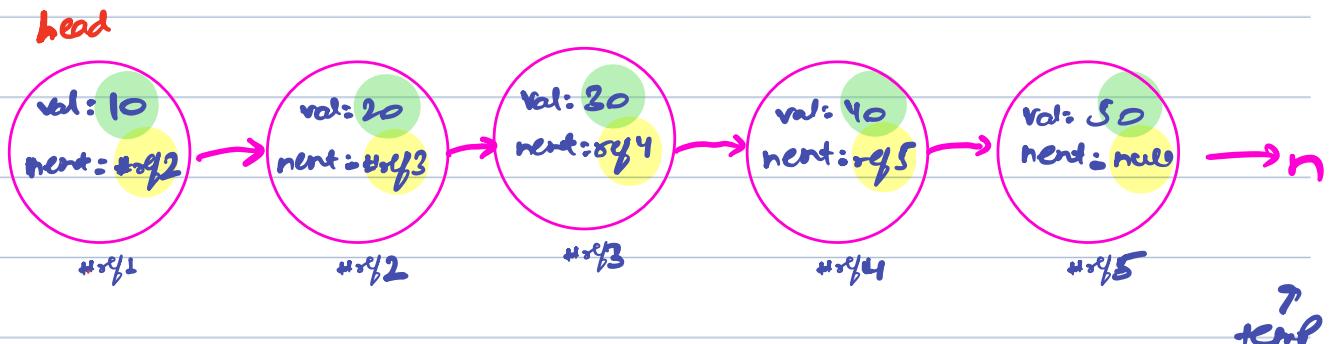
```
    val = v;
```

```
}
```



Q) Point linkedlist

↳ Given head of the LL Point the elements.



10 20 30 40 50

void Pointlinkedlist (node head){

 node temp = head;

 while (temp != null) {

 Point (temp.val);

 temp = temp.next;

T.C.: O(n)

S.C.: O(1)

+temp.next = temp;

It updates the

next arrow.

3

for (node temp ; temp != null ; temp = temp.next) {

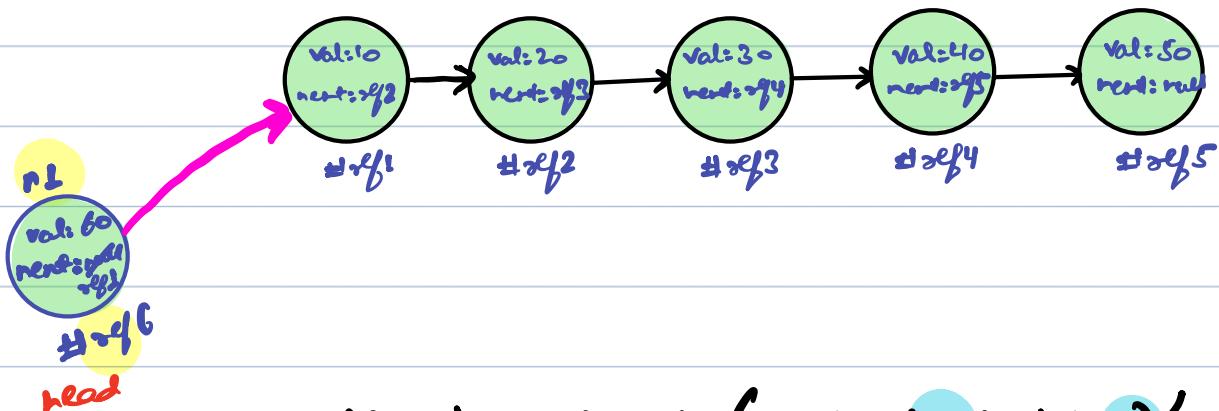
3



Q) insert in a linkedlist

↳ insert at head

v=60



void insertathead (node head, int v) {

 node n1 = new node (v);

 n1.next = head;

 head = n1;

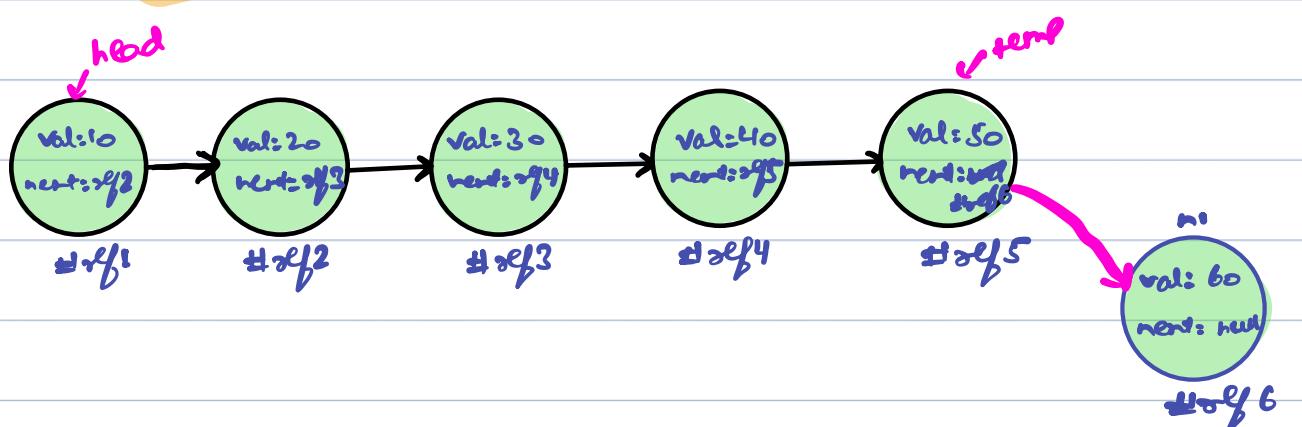
T.C: O(1)

S.C: O(1)

3



insert after last node
↳ Y = 60



void insertatend (node head, int v) {

 node n1 = new node (v);

 node temp = head;

 while (temp.next != null) {

 temp = temp.next;

 temp.next = n1;

}

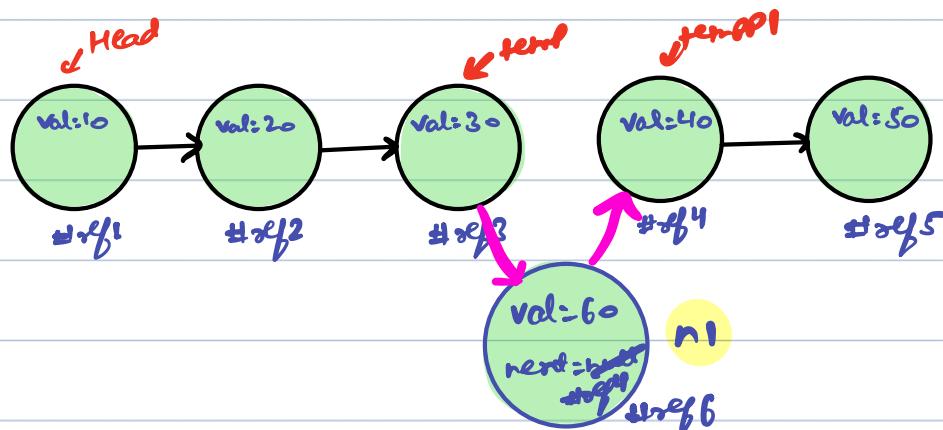
T.C: O(n)

S.C: O(1)



if insert at index $\rightarrow K=3, v=60$

head
+
null



void insertat(node head, int k, int v){

node n1 = new node(v);

Node temp = head;

for(int i=1; i<=K-1; i++) {

temp = temp.next;

node tempPL = temp.next;

temp.next = n1;

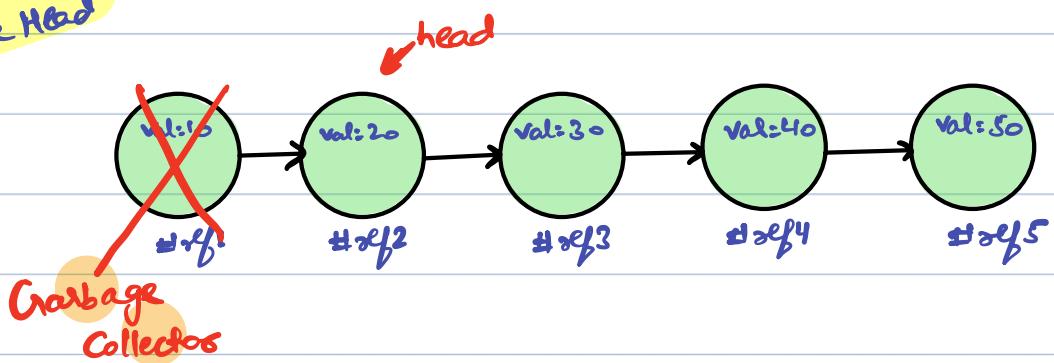
n1.next = tempPL;

3



Q) Delete in a linkedlist

→ delete Head



T.C: O(1)

↳ `head = head.next;`

S.C: O(1)

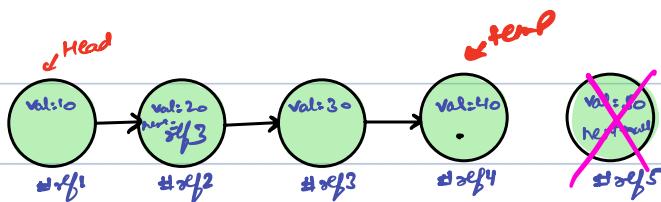


AlgoPrep

Break till 10:40 pm



→ delete last



T.C: $O(n)$

S.C: $O(1)$

`deletelast (Node head) {`

`Node temp = head;`

`while (temp.next.next != null) {`

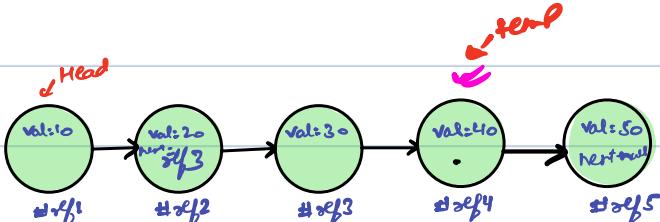
`temp = temp.next;`

`temp.next = null;`

Tracing

`while (temp.next.next != null) {`

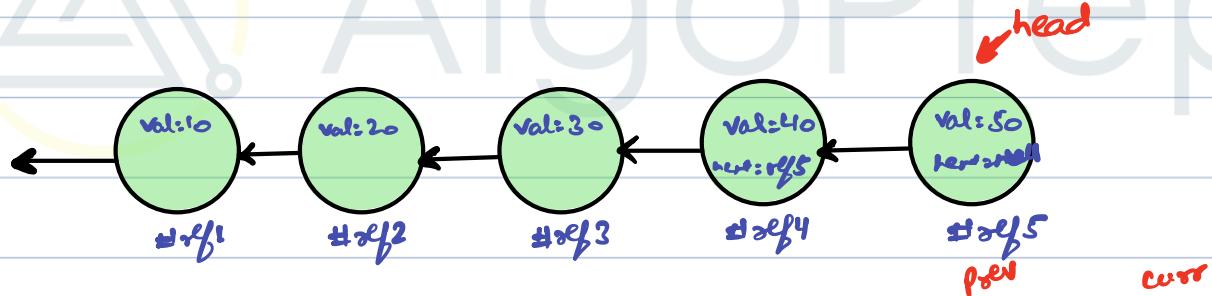
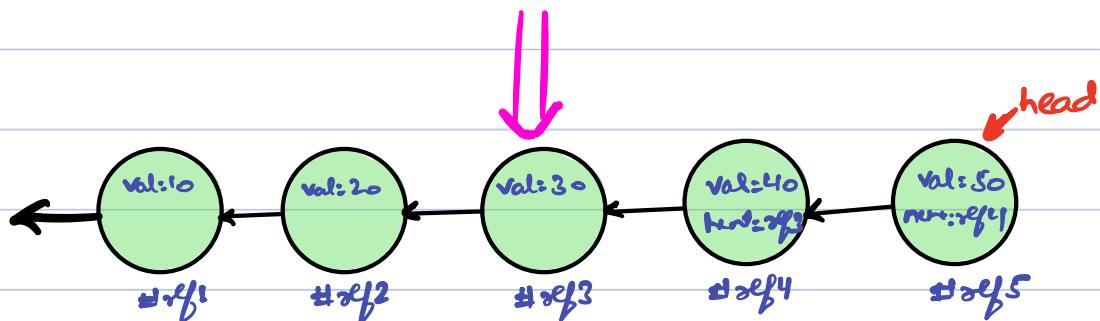
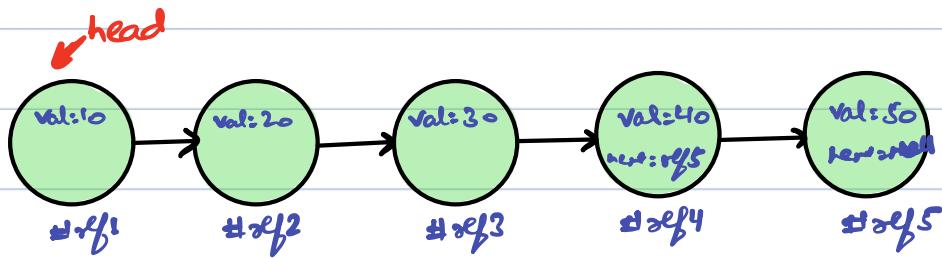
`temp = temp.next;`



`head.next = null`



Q) Reverse a linkedlist



`Node curr = head;`
`Node prev = null;`

`while (curr != null) {`

`Node currNext = curr.next;`

`curr.next = prev;`

`prev = curr;`

`curr = currNext;`

}

`head = prev;`

T.C: $O(n)$

S.C: $O(1)$



Q) Find mid



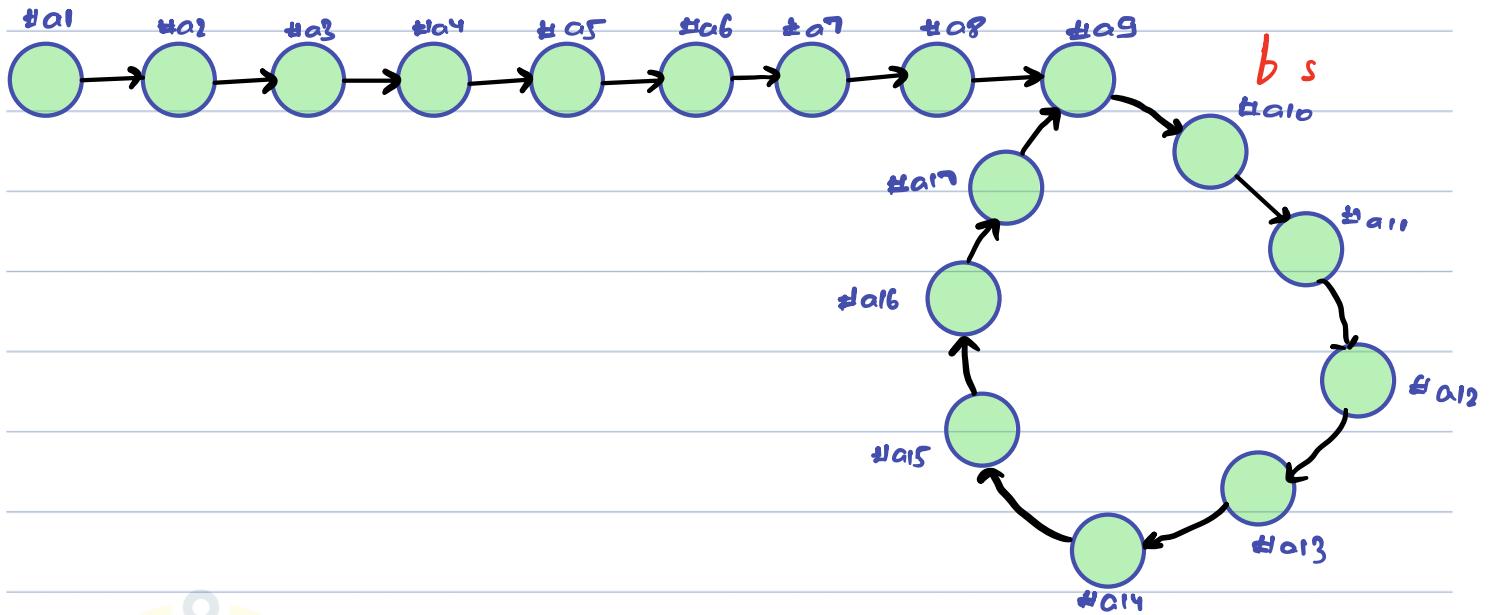
AlgoPrep

floyd cycle

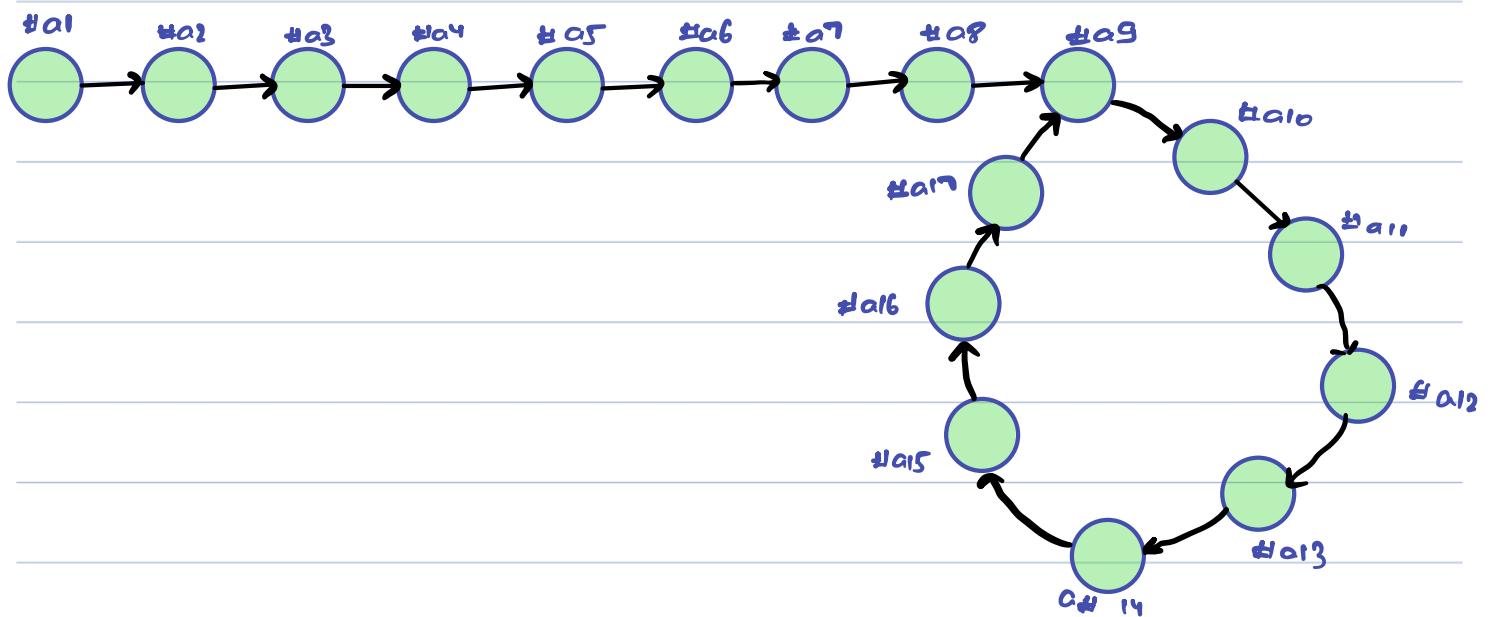


Q) Given a linked list, Check for cycle & return the Starting Point if exists.

head

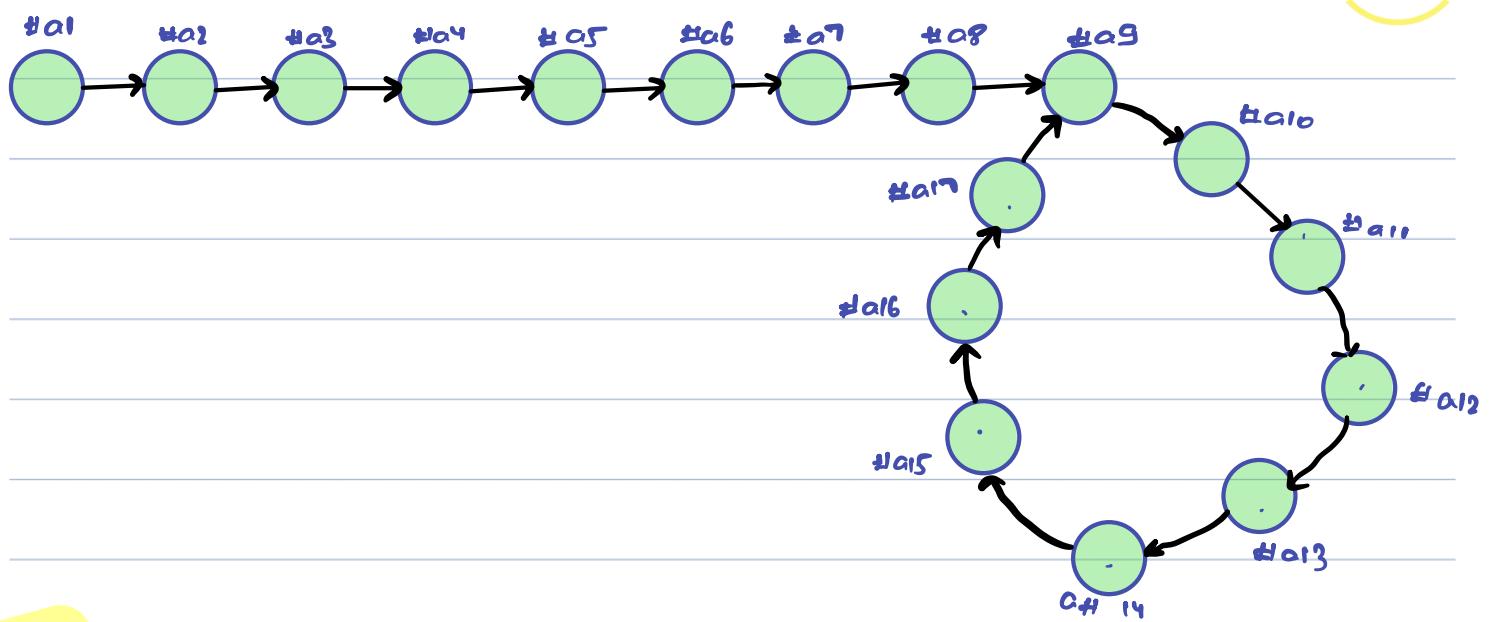


head



End:

head



End:

Chain = 4

Cycle = 9

