Today's agenda

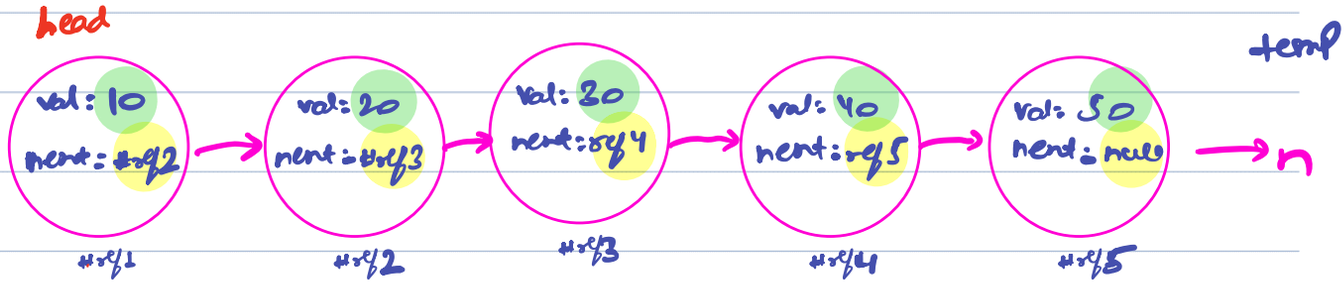
- Print linkedlist
- insert in a linkedlist
- delete in a linkedlist
- Reverse the linkedlist
- Find mid
- floyd cycle.

//Nested Class

```
Class Node {
    int val;
    Node next;

    Node (int v1){
        val = v1;
    }
}
```

Q) Print LinkedList

Given head of the LL Print the elements.



10 20 30 40 50

10 20 30 40 50

```
void PrintLinkedList (Node head){
    Node temp = head;
    While (temp != null) {
        S.O.P (temp.val);
        temp = temp.next;
    }
}
```

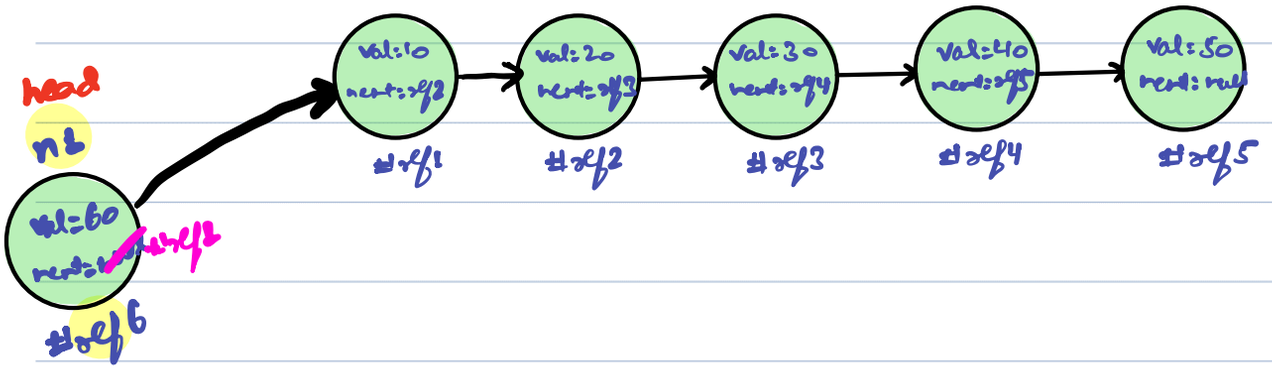
T.C: O(N)

S.C: O(1)

Q) insert in a linkedlist

↳ insert at head

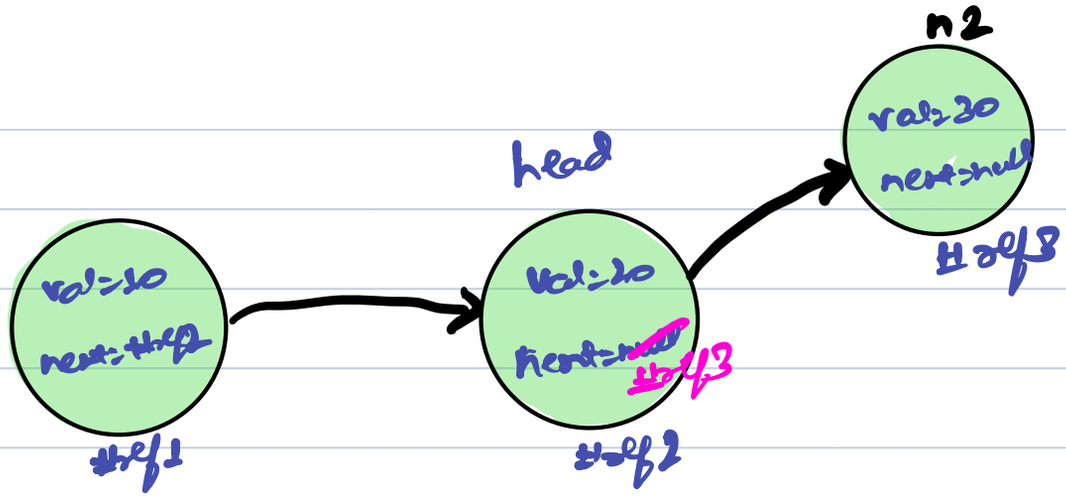
v = 60




```
void insertHead (Node head, int v){

    Node nl = new Node(v);

    nl.next = head;

    head = nl;

}
```

T.C: O(1)

S.C: O(1)



`head = head.next;` → i = i+1

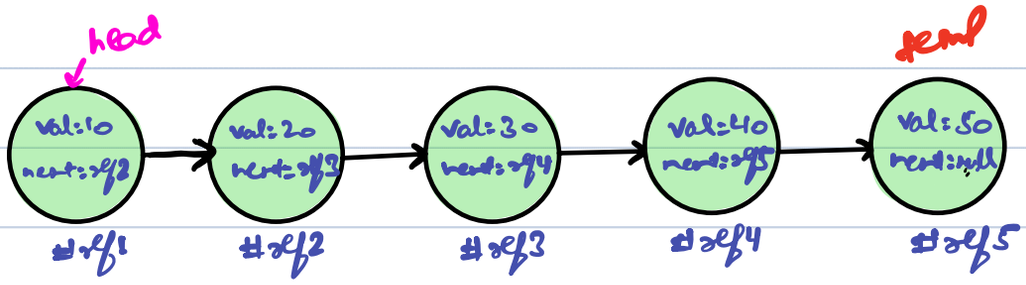
↳ head jumps to next.

`head.next = n2;`

↳ Creates a connection

insert after last node

↳ y = 60

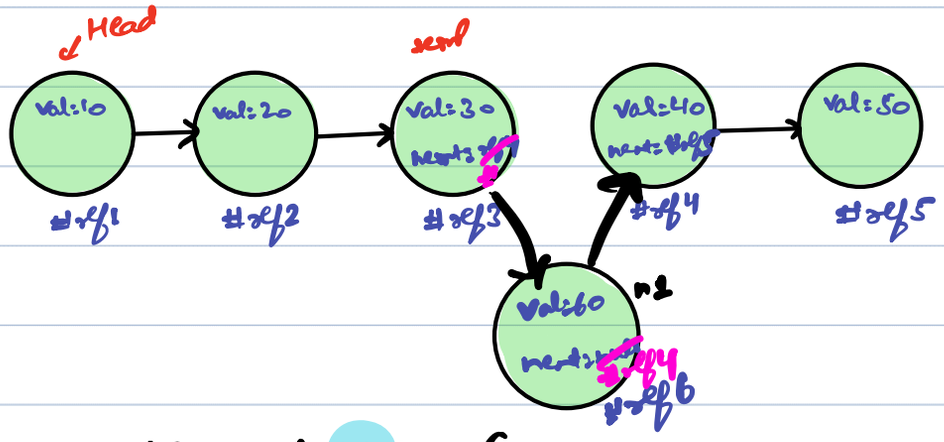



```
void insertatend(Node head, int v){

    Node nl = new Node(v);

    Node temp = head;

    while(temp.next!=null){
        temp = temp.next;
    }

    temp.next = nl;

}
```

T.C: O(N)

S.C: O(1)

// insert at index → K=3, V=60



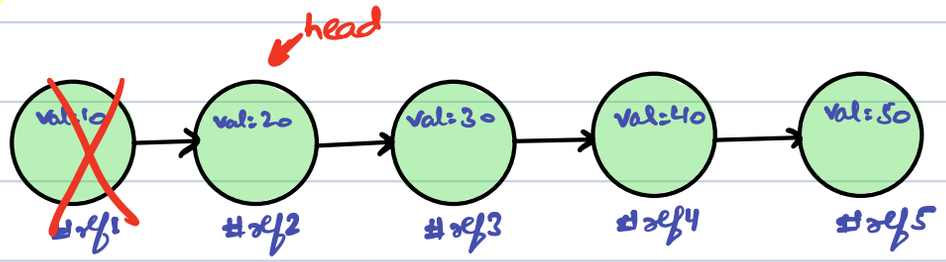

```
void insertat (Node head, int v) {

    Node n1 = new Node(v);

    Node temp = head;

    for (int i=1; i<=k-1; i++) {
        temp = temp.next;
    }

    Node temp1 = temp.next;

    temp.next = n1;

    n1.next = temp1;

}
```

T.C: O(K) = O(N)

S.C: O(1)

Q) Delete in a linkedlist

→ delete Head



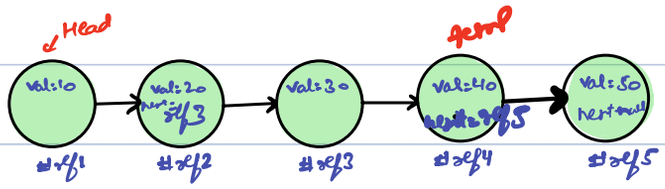
Garbage collector

↳ `head = head.next;`

T.C: O(1)

S.C: O(1)

→ delete last




```
void deleteLast (Node head){

    Node temp = head;

    while (temp.next.next!=null){
        temp = temp.next;
    }

    temp.next = null;

}
```

T.C: O(N)

S.C: O(1)

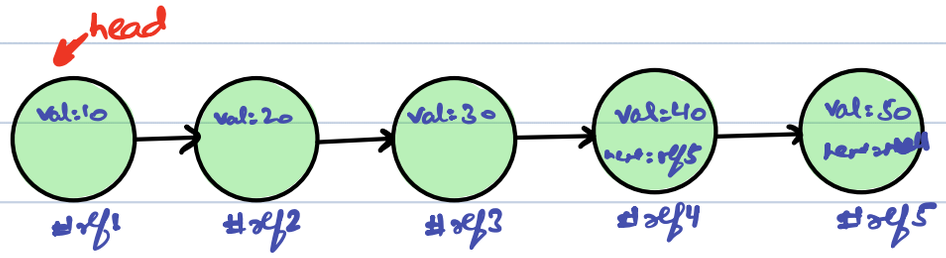
→ edge cases

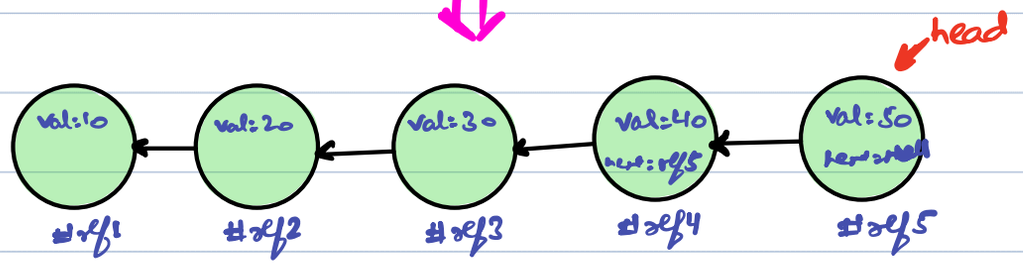
↓

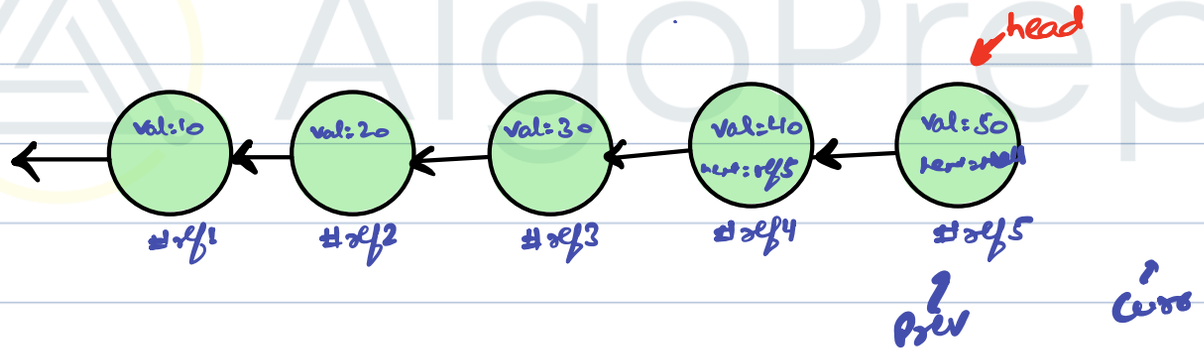
H.W

Break till 10:35 Pm

Q) Reverse a linkedlist








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

while (curr != null) {
    Node currP1 = curr.next;
    curr.next = prev;
    prev = curr;
    curr = currP1;
}

head = prev;
```

T.C: O(N)

S.C: O(1)