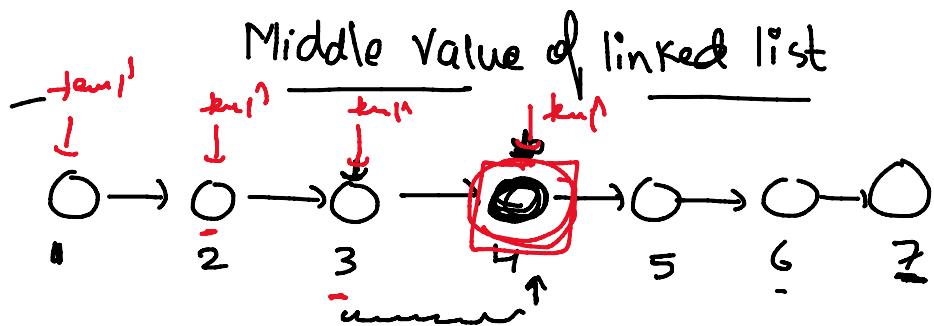


CLASS - 51

Find the Middle Node of linked list

Brute Force - Calculate length of linked list = $n = 7$

Middle Node \rightarrow

$$\frac{G+1}{2} = \frac{7+1}{2} = 4$$

$$\frac{(n+1)}{2}$$

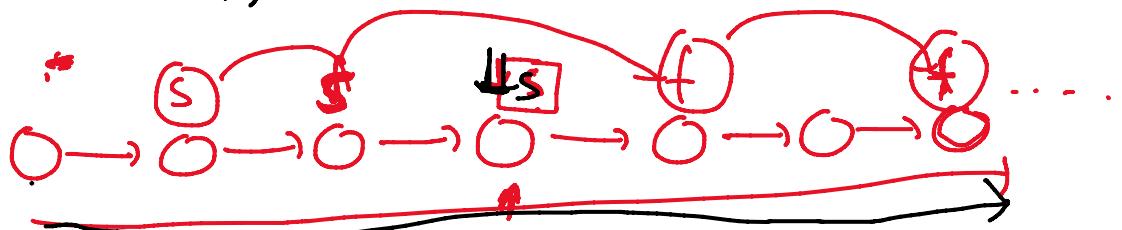
$$\frac{7+1}{2} = \frac{8}{2} = 4$$

$\text{Node} * \text{temp} = \text{head};$

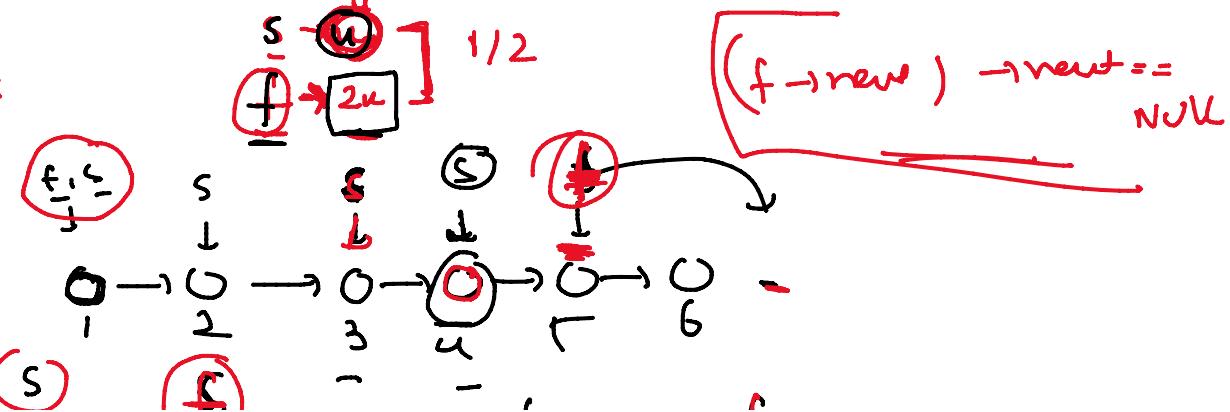
$$\text{int idm} = \left(\frac{n+1}{2} \right)$$

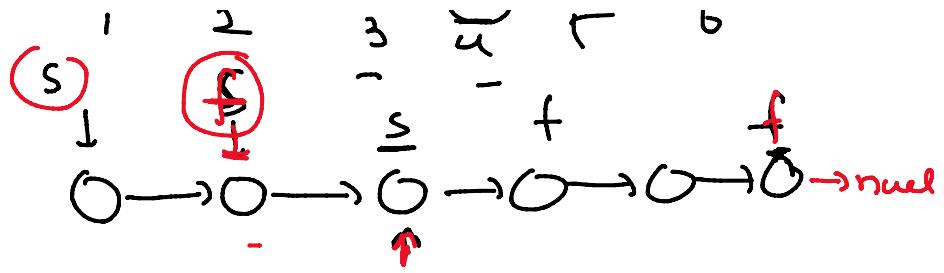
```
for ( i=1; i< idm; i++ )
    if ( temp == temp )  $\rightarrow$  next;
    ...
return temp;
```

2nd Approach



Fast Pointer
Slow Pointer





(fast → next) → next

Node * s = Head;

Node * f = Head;

while (f != NULL) and f → next != NULL)

f = f → next → next;

s = f → next;

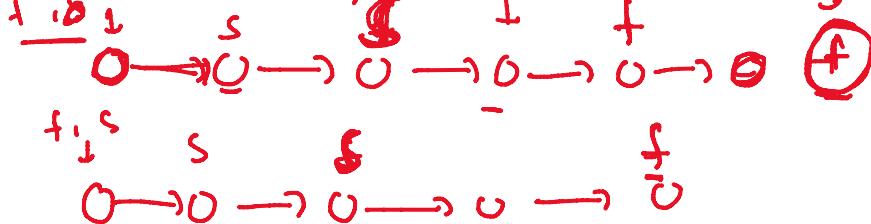
f = NULL

s = u

γ

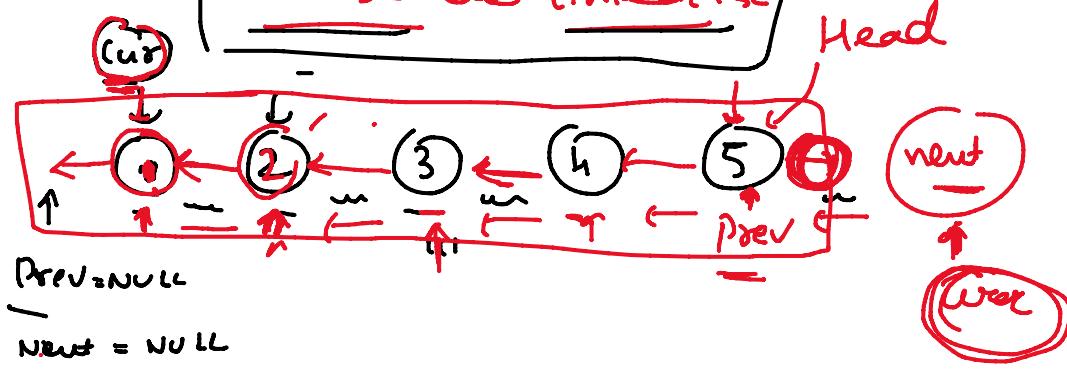
return (s);

s



length

Reverse as linked list



Node * (cur = head);

```

Node** cur = head;
Node* prev = NULL;
Node* next = NULL;

```

```
while (current != NULL)
```

```

    ↪ next = (cur->next);
    (cur->next = prev);
    prev = current;
    ↴
    cur = next;

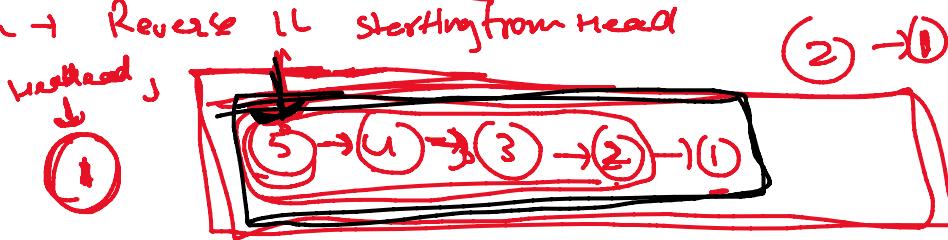
```

Head = prev

How to Reverse using Recursion



Problem → Reverse LL starting from Head



Head -> next -> next = Head

Head -> next = NULL;

CP

Node * reverse (Node * Head)

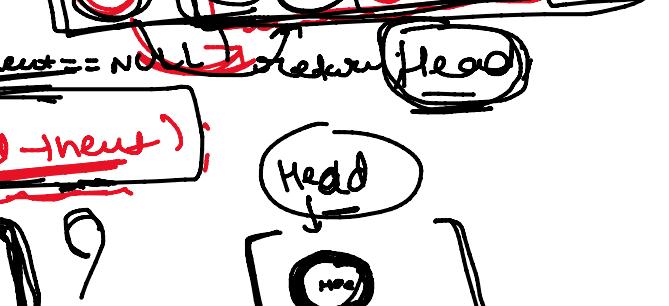
if (Head == NULL OR Head->next == NULL) return Head;

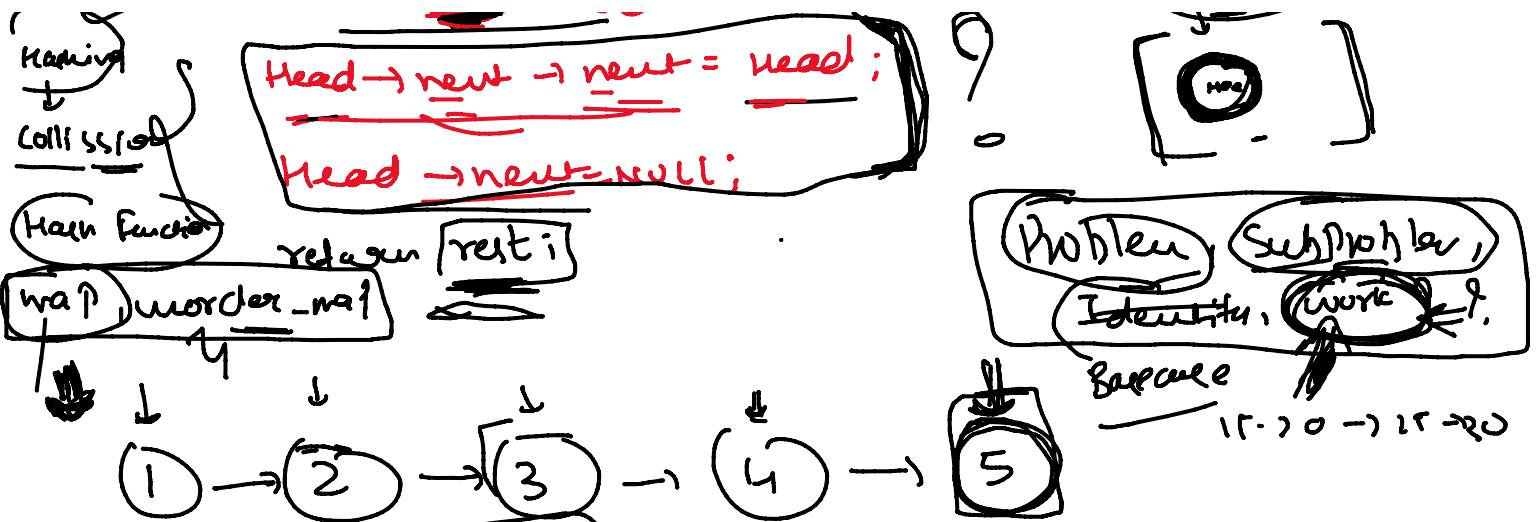
Node * rest = reverse (Head->next);

Head->next->next = Head;

Med
concept
store

return
+





Reverse
 using linked list

reverse (1)

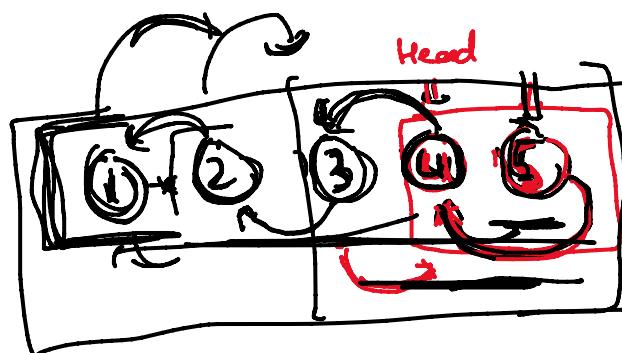
reverse (2)

reverse (3) rest=5

reverse (4)

rest=5

reverse (5)



Node & reverse

CP >>>

Projects

Grat

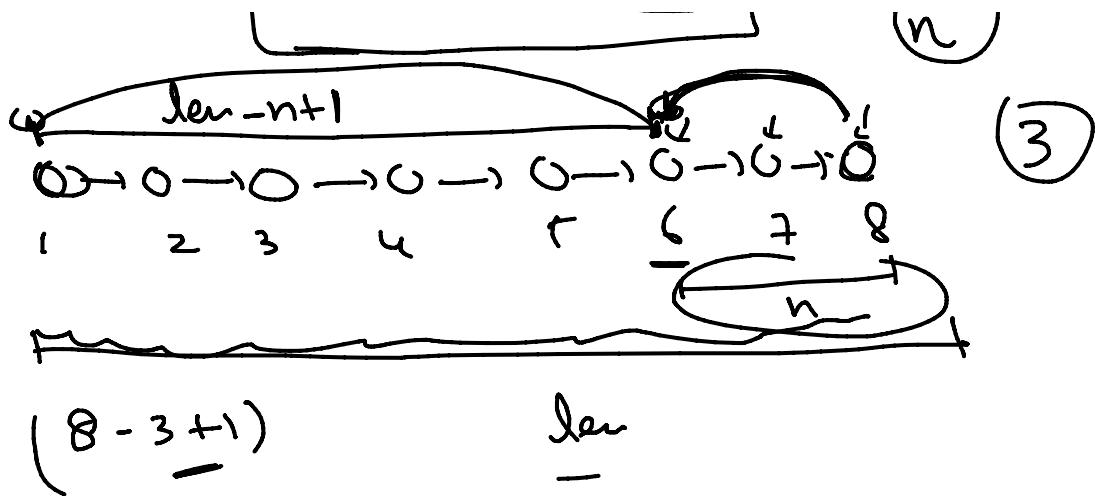
(1g, Gador, Profek)

OA

= link

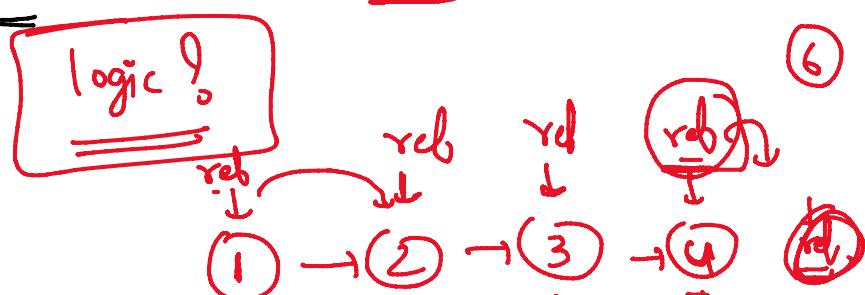
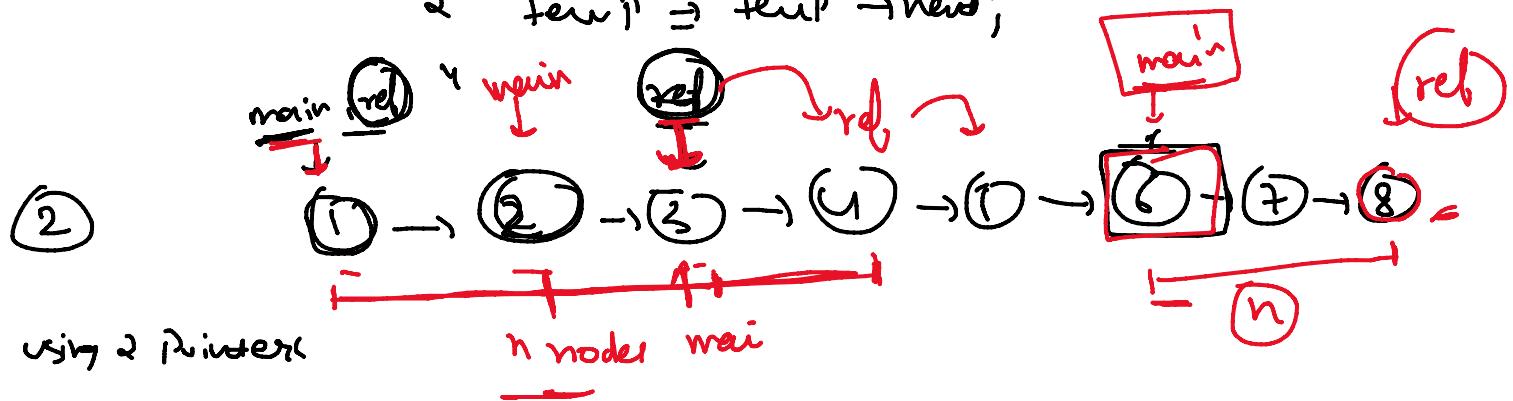
N Node from last

n



```
for (i=1; i<(len-n+1); i++)
```

```
{ temp  $\doteq$  temp->next;
```



```
for (i=1; i<5; i++)
```

```
{ ref = ref->next;
```

```
if (ref == NULL)
```

```
{ return; }
```

$n > \text{len}$

Intersection of 2 linked list

