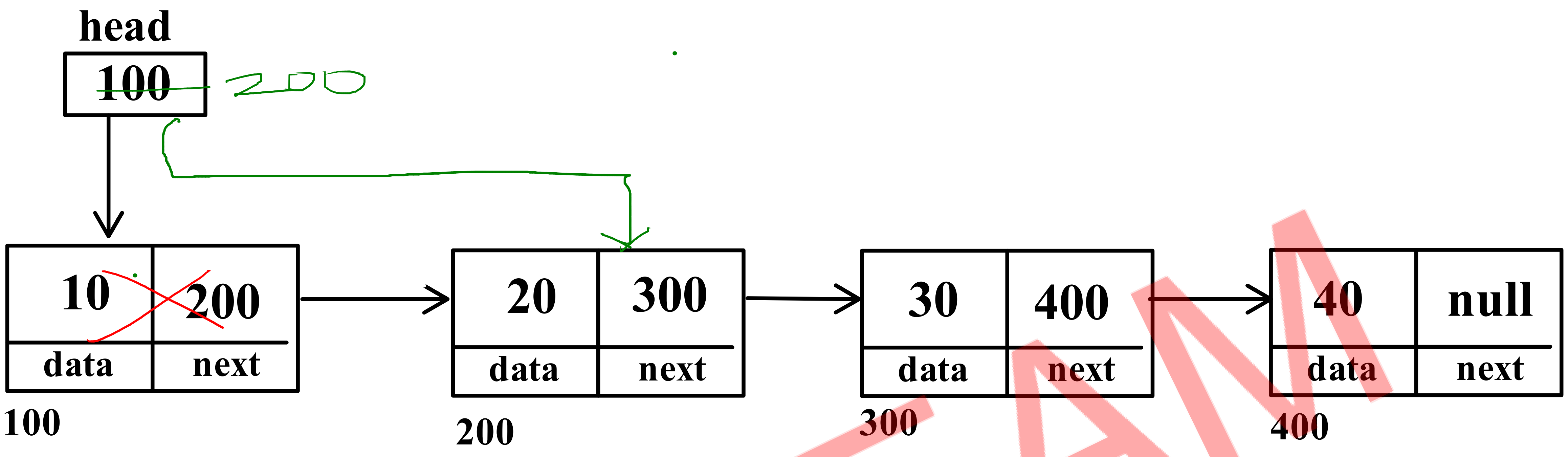
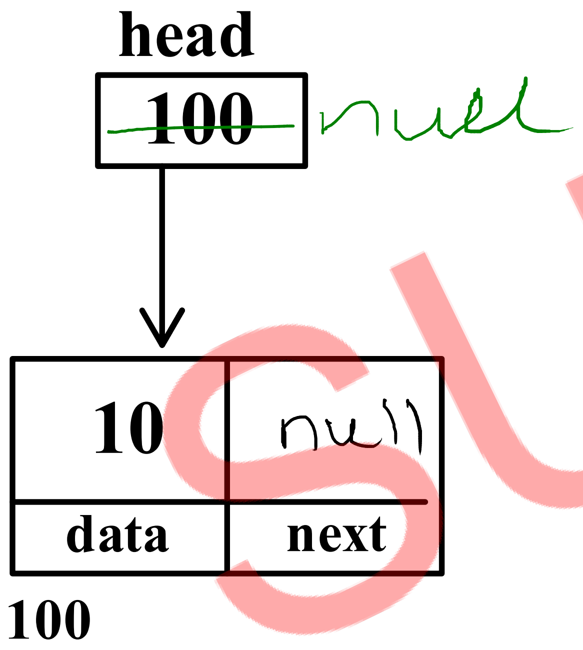
Singly Linear Linked List - Delete First



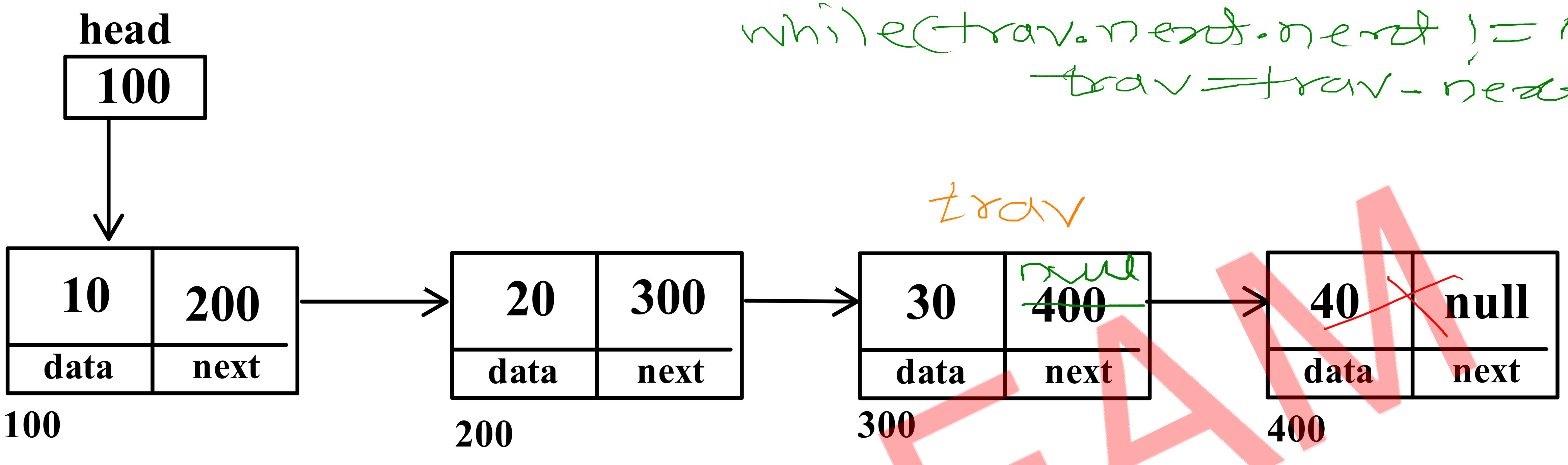


```
//1. if list is empty
        return ;
//2. if list is not empty
        //a. move head on second node
```

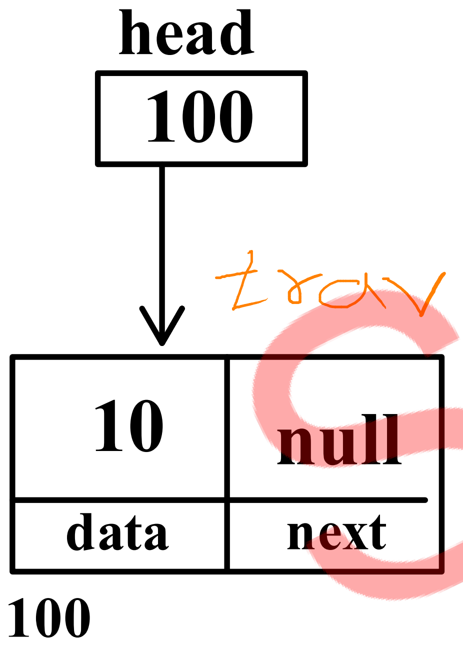
$T(n) = O(1)$

head=head.next

Singly Linear Linked List - Delete Last



```
while(trav.next.next != null)
    trav = trav.next;
```



//1. if list is empty

    return;

//2. if list has single node

    head = null;

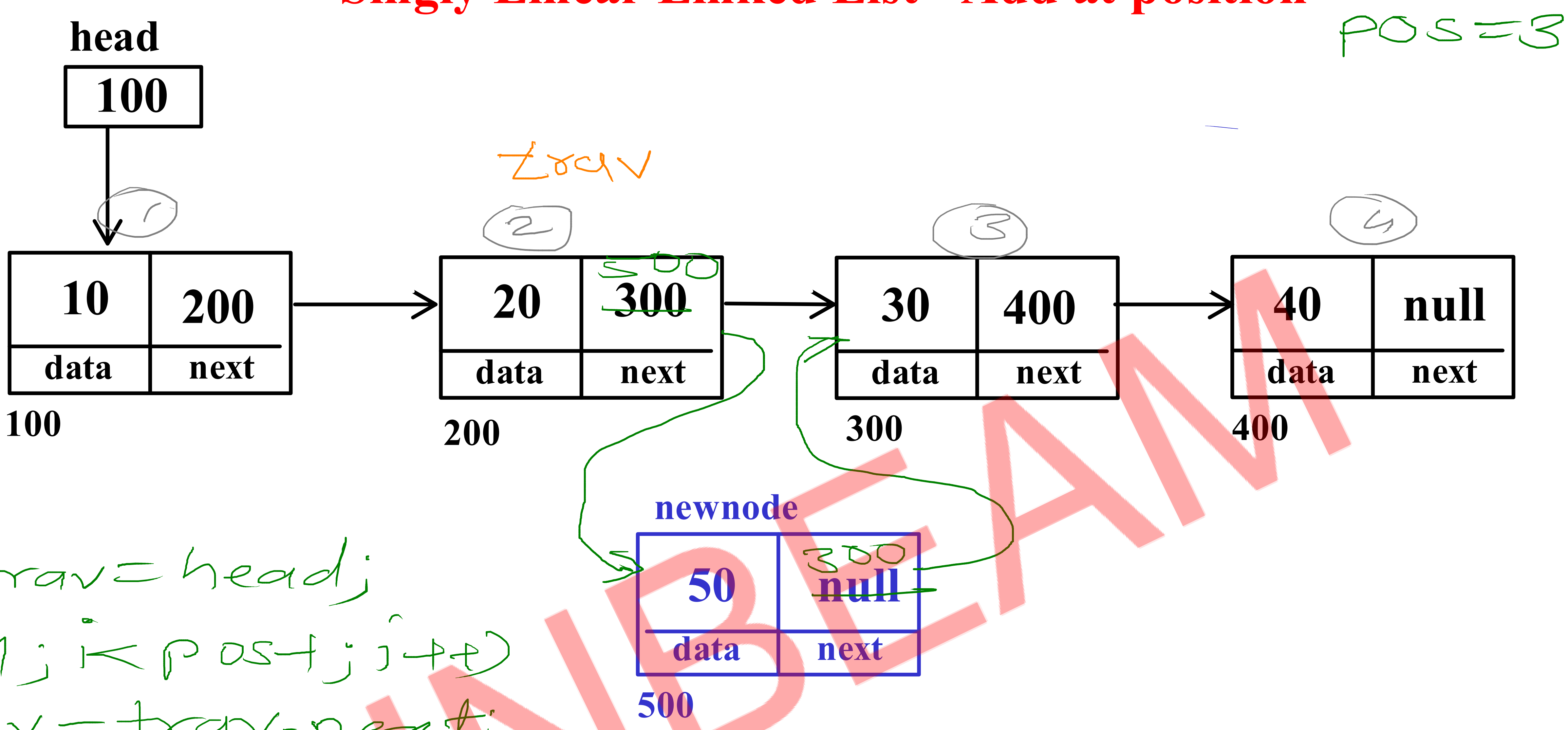
//3. if list has multiple nodes

    //a. traverse till second last node

    //b. make next of second last node null

$T(n) = O(n)$

Singly Linear Linked List - Add at position



```
Node trav=head;
for(i=1; i<pos; i++)
    trav=trav.next;
```

pos=3

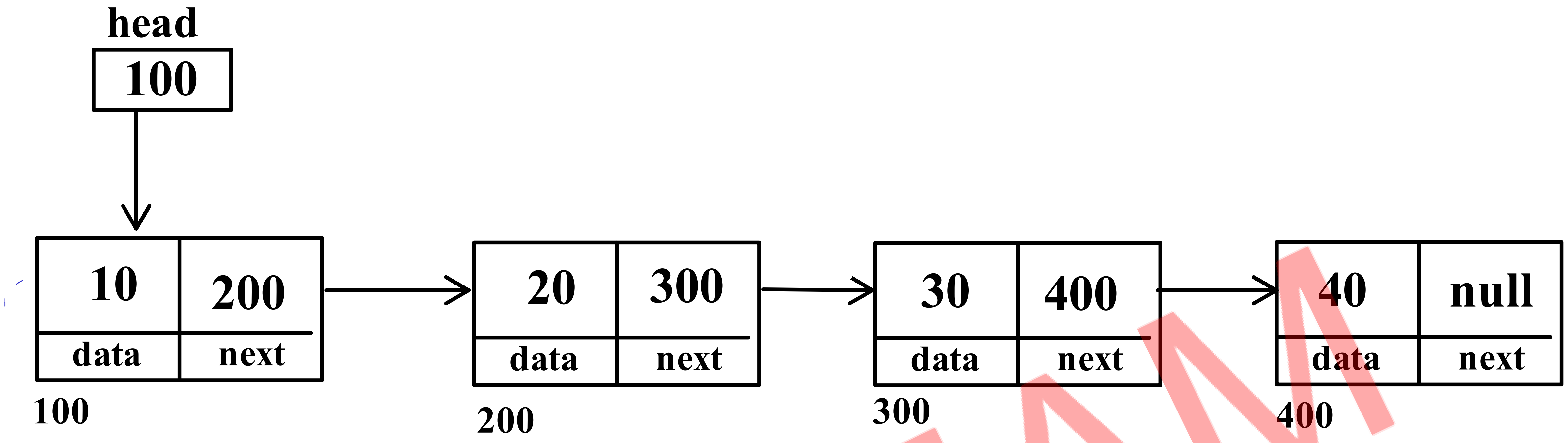
trav	i	i<2
100	1	T
$\boxed{200}$	2	F

pos=4

trav	i	i<3
100	1	T
200	2	T
300	3	F

```
//1. create node with given data
//2. if list is empty
        // add newnode into head itself
//3. if list is not empty
        //a. traverse till pos - 1 node
        //b. add pos node into next of newnode
        //c. add newnode into next of pos-1 node
```

$T(n) = O(n)$



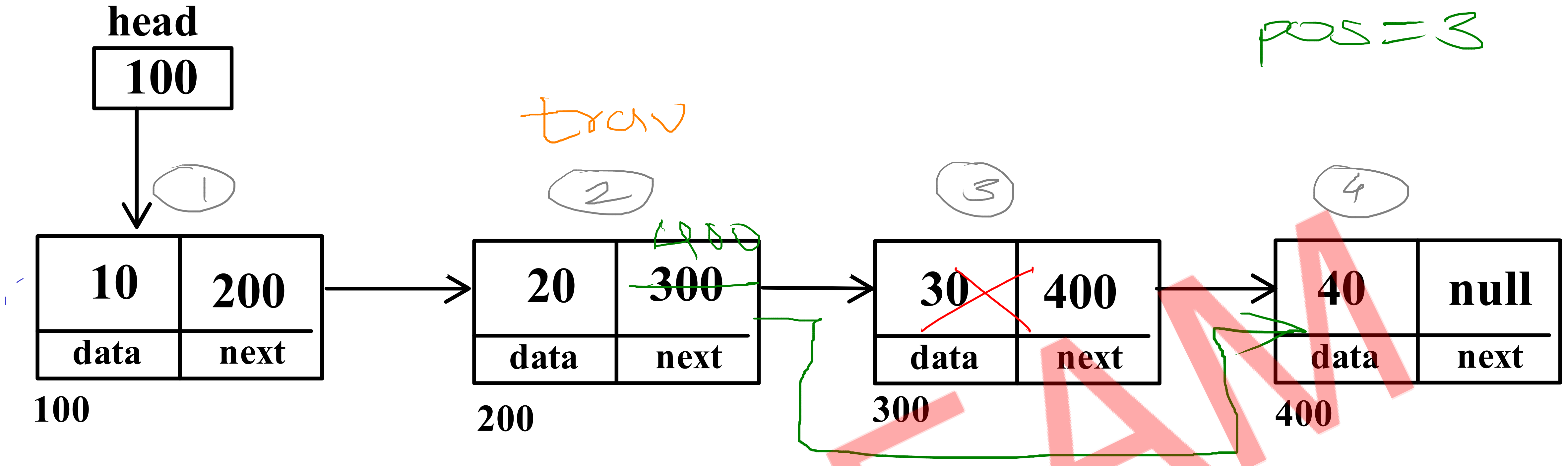
```
Node trav=head;
for(i=1; i<pos-1;i++)
    trav=trav.next;

newnode.next=trav.next;
trav.next = newnode;
```

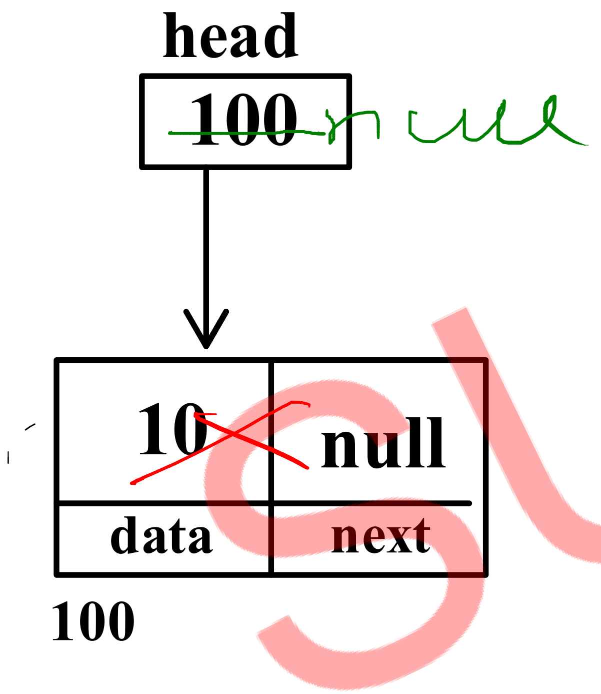
pos=6

trav	i	i<5
100	1	T
200	2	T
300	3	T
400	4	T
null	5	F

Singly Linear Linked List - Delete at position

pos=3

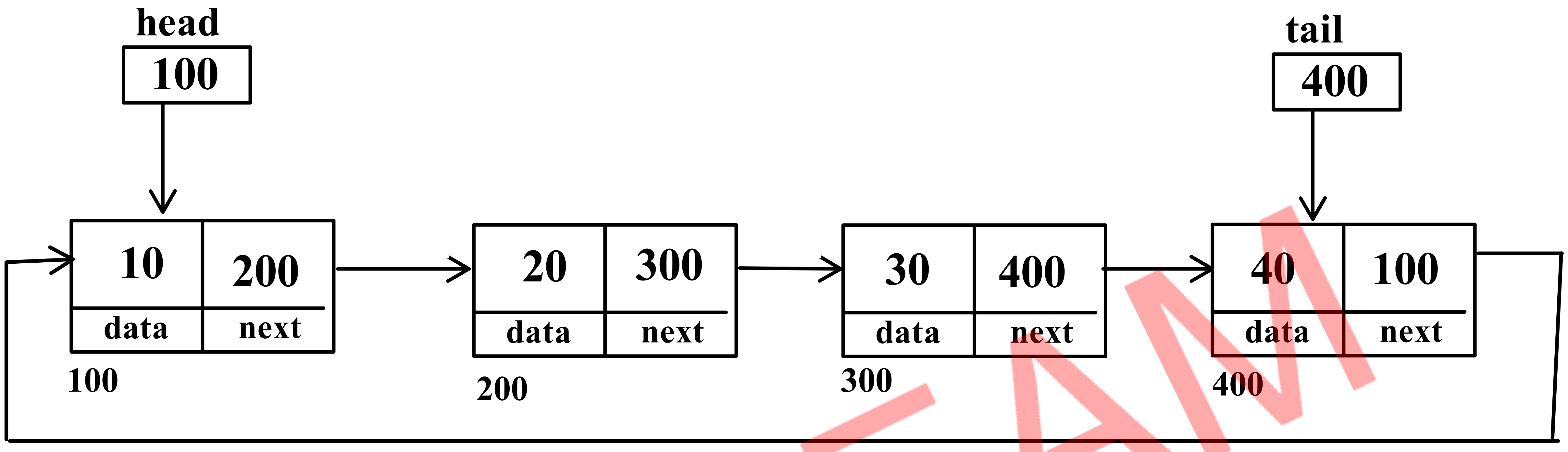




```
//1. if list is empty
        return;
//2. if list has single node
        head = null;
//3. if list has multiple nodes
        //a. traverse till pos-1 node
        //b. add pos+1 node into next of pos-1 node
```

T(n)=O(n)

Singly Circular Linked List - Display



```
Node trav = head;
do
{
    sysout(trav.data)
    trav = trav.next;
} while (trav != head)
```

$T(n) = O(n)$

//1. create trav and start at head

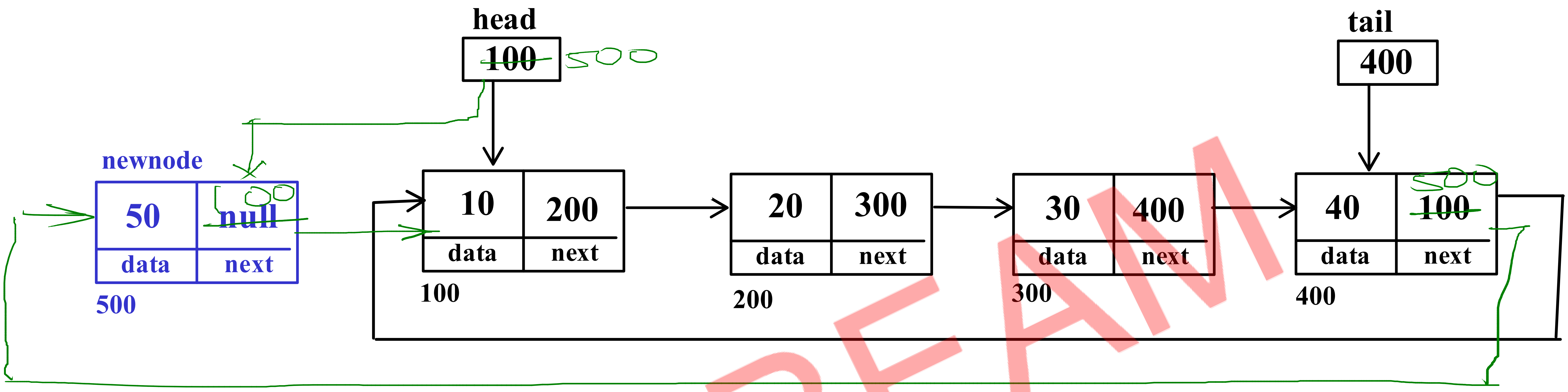
//2. print data of current node

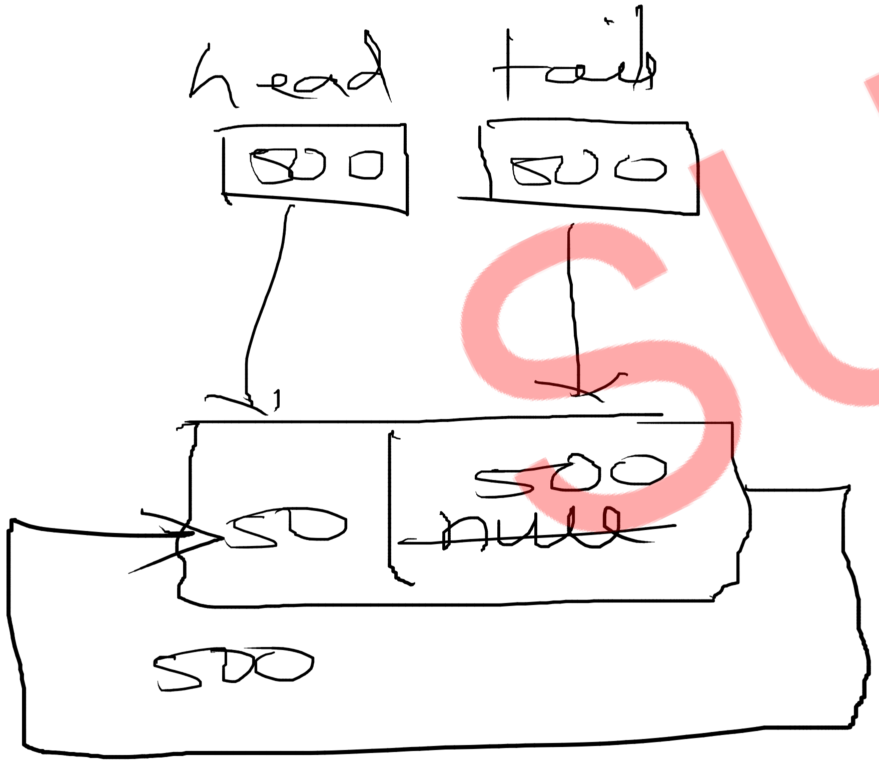
//3. go on next node

//4. repeat step 2 and 3 till last node

trav	trav.data
100	10
200 ✓	20
300 ✓	30
400 ✓	40
100 ✗	

Singly Circular Linked List - Add First





//1. create node with value

//2. if list is empty

    //a. add newnode into head and tail

    //b. make list circular

//3. if list is not empty

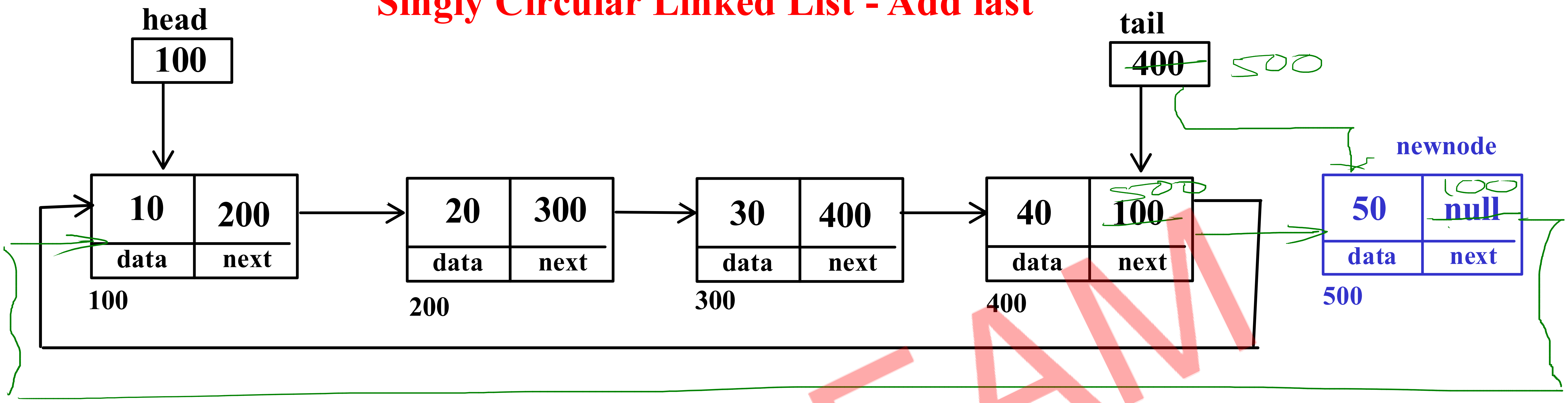
    //a. add first node into next of newnode

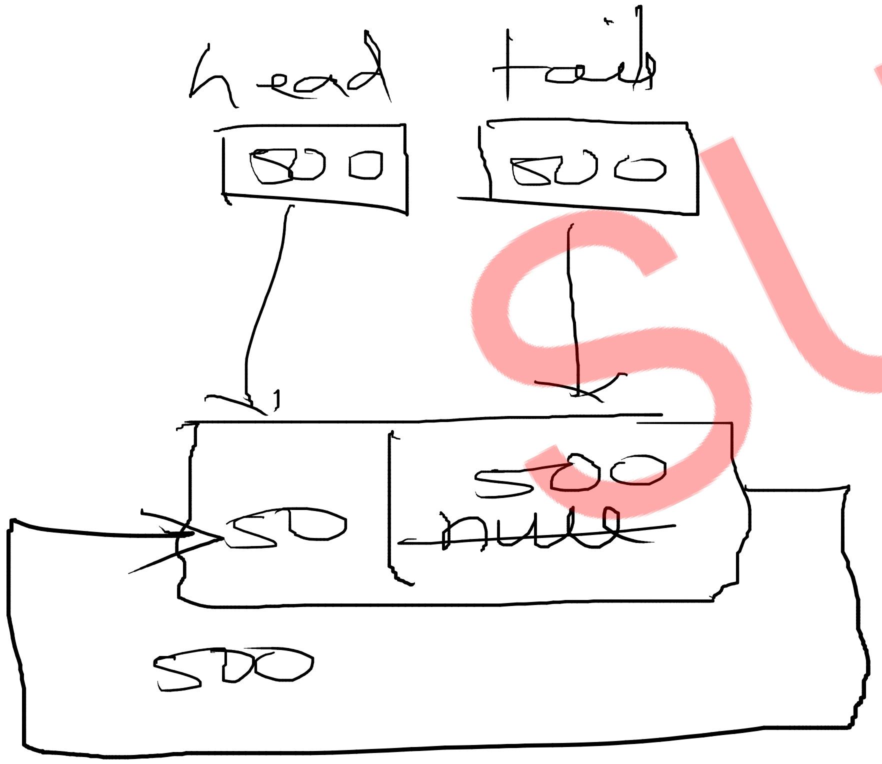
    //b. add newnode into next of last node

    //c. move head on newnode

T(n) = O(1)

Singly Circular Linked List - Add last

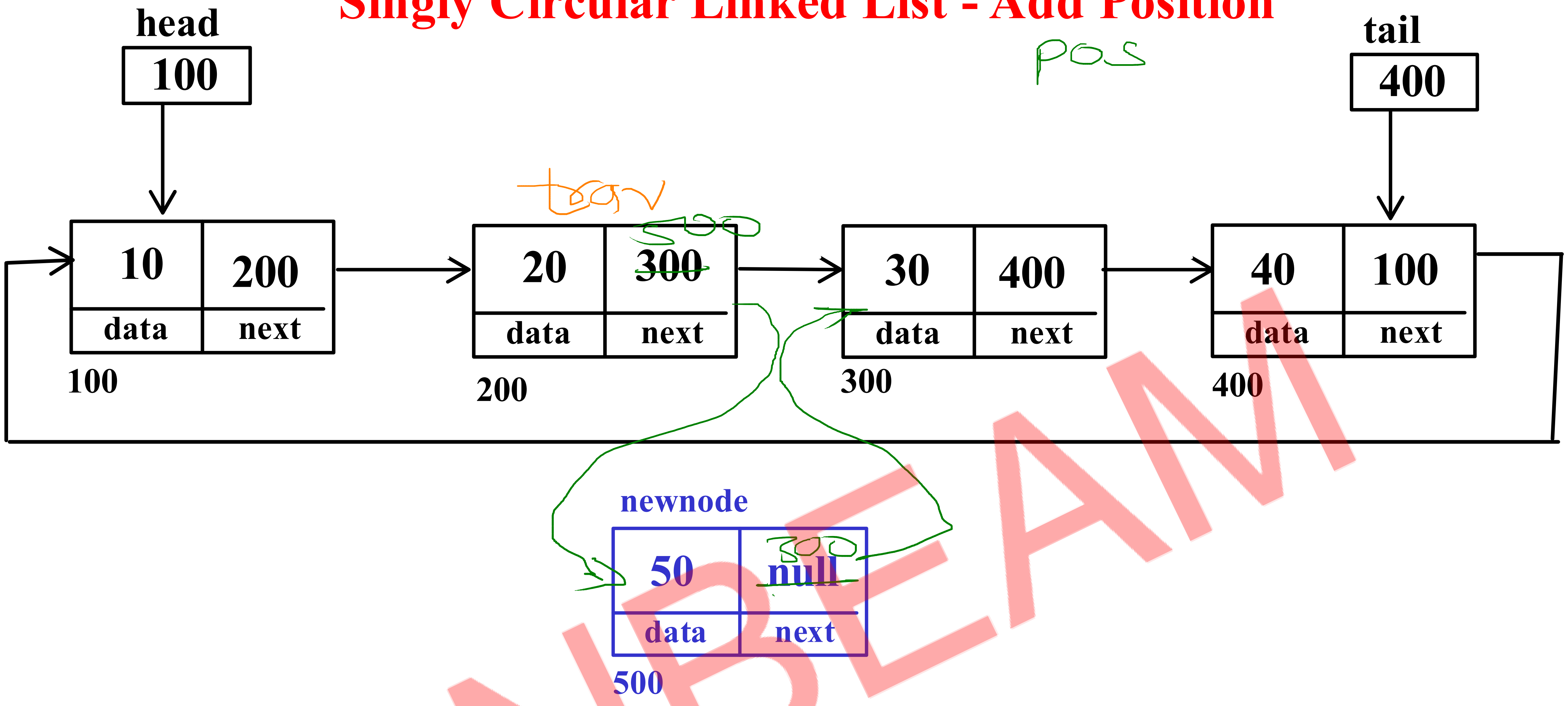




```
//1. create node with value
//2. if list is empty
        //a. add newnode into head and tail
        //b. make list circular
//3. if list is not empty
        //a. add first node into next of newnode
        //b. add newnode into next of last node
        //c. move tail on newnode
```

T(n) = O(1)

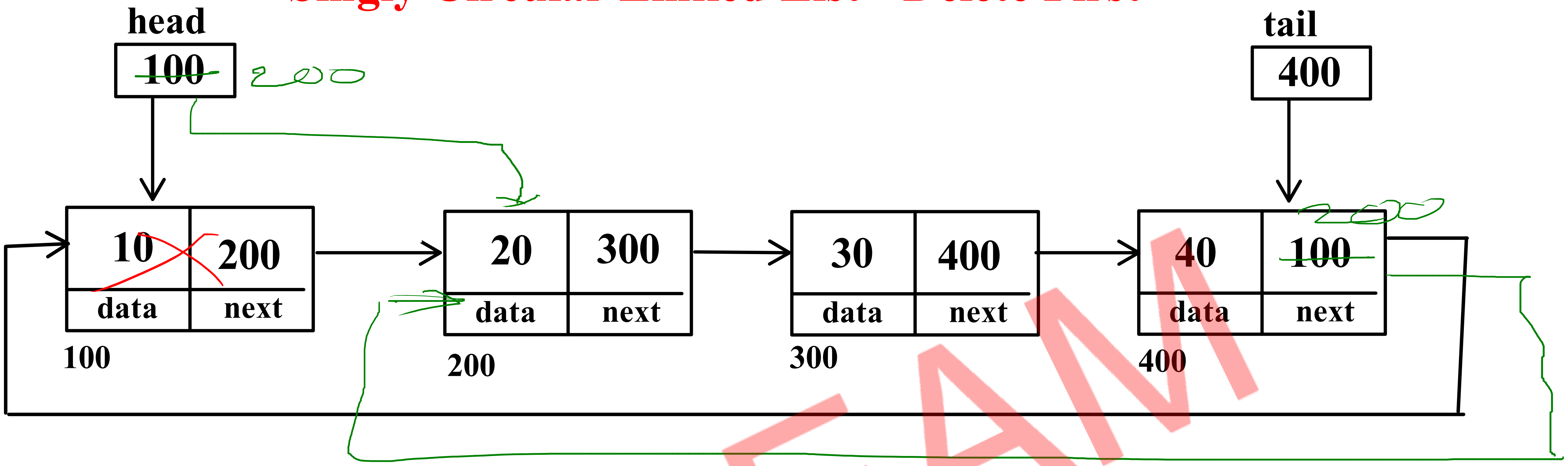
Singly Circular Linked List - Add Position

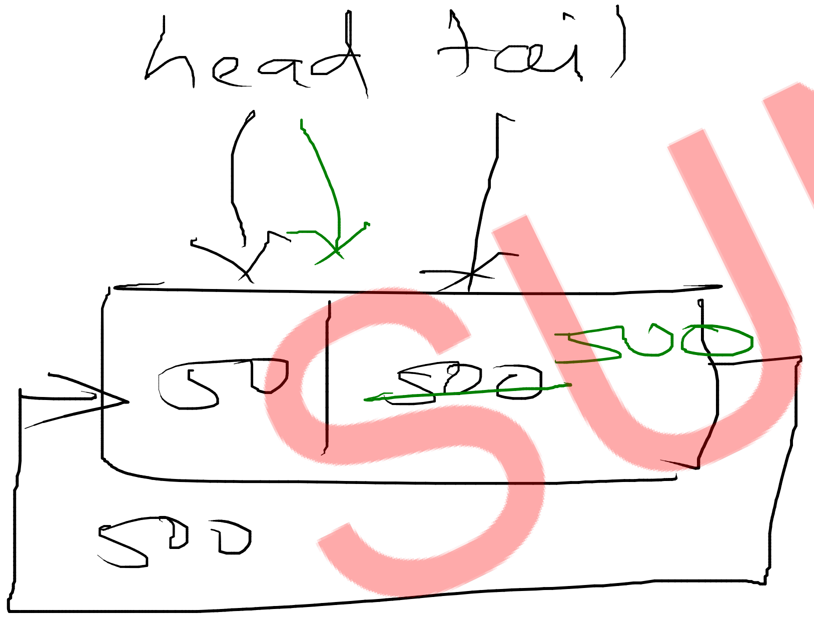


```
//1. create node with given data
//2. if list is empty
        // add newnode into head and tail itself
        // make list circular
//3. if list is not empty
        //a. traverse till pos - 1 node
        //b. add pos node into next of newnode
        //c. add newnode into next of pos-1 node
```

T(n) = O(n)

Singly Circular Linked List - Delete First





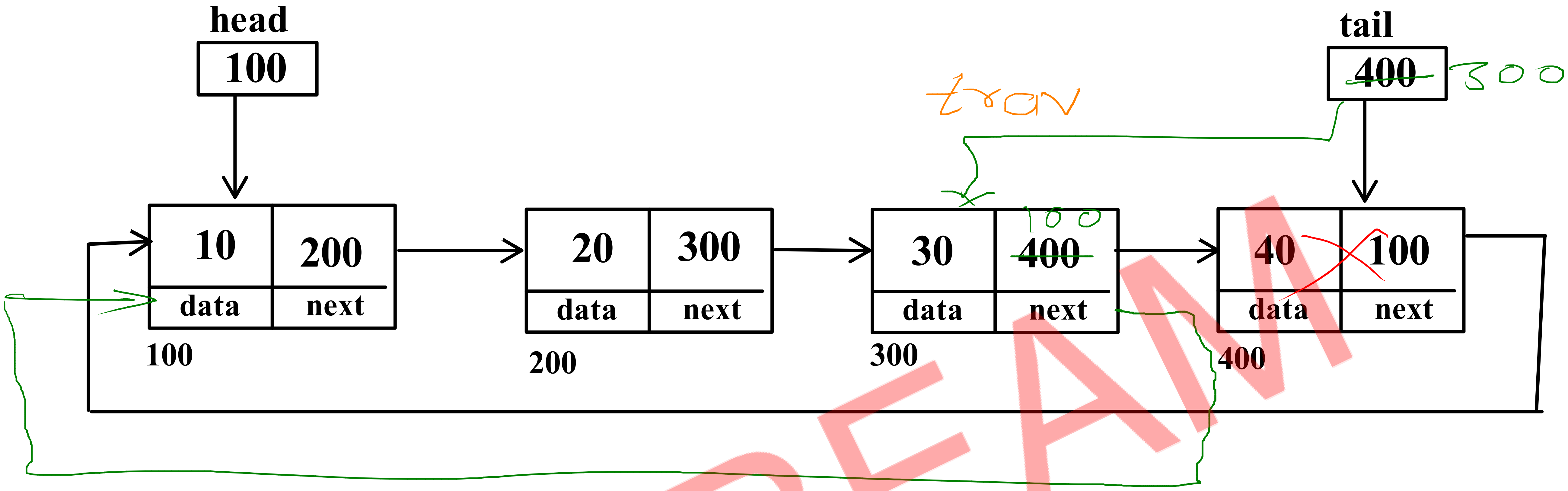
```
//1. if list is empty
        return;
//2. if list has single node
        head = tail = null;
//3. if list has multiple nodes
        //a. add second node into next of last node
        //b. move head on second node
```

tail.next = head.next

head = head.next

T(n) = O(1)

Singly Circular Linked List - Delete last



```
//1. if list is empty
        return;
//2. if list has single node
        head = tail = null
//3. if list has multiple nodes
        //a. traverse till second last node
        //b. add first node into next of second last node
        //c. move tail on second last node
```

T(n) = O(n)