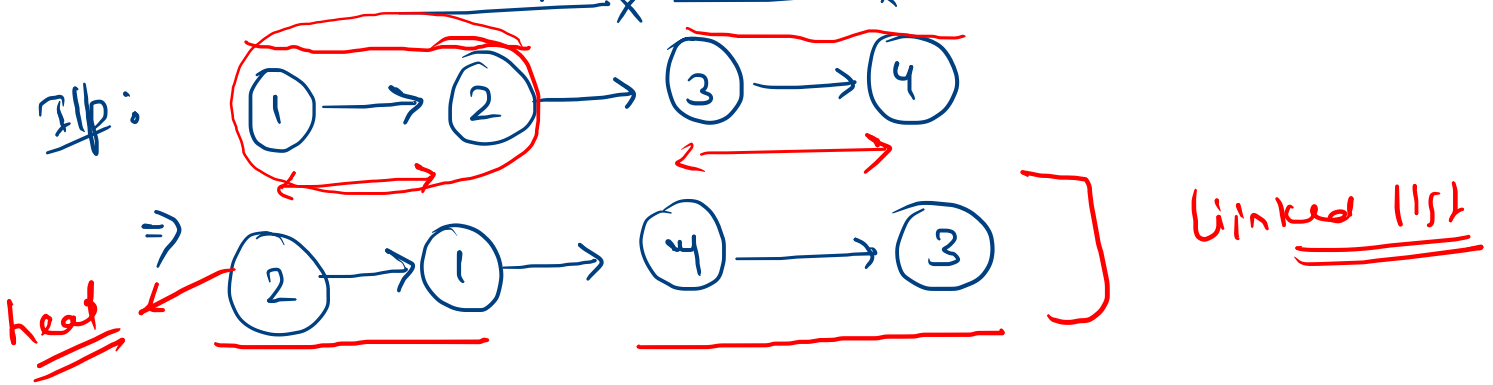
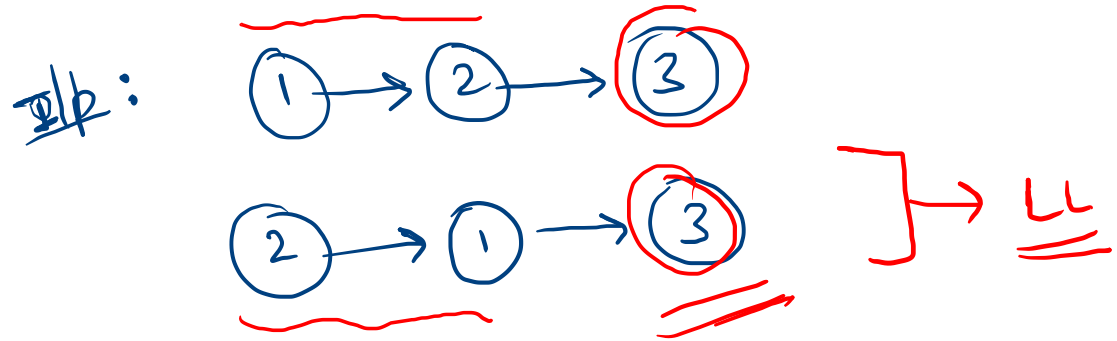
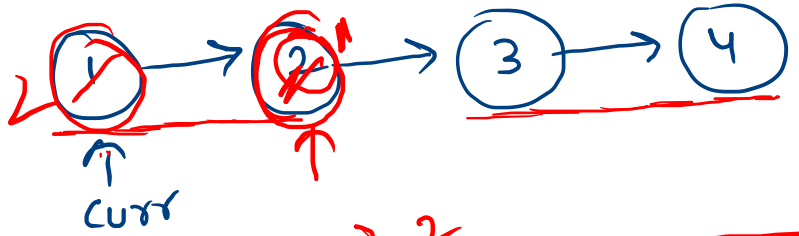
Swap Nodes in Pair





NAIVE Approach

I/p :

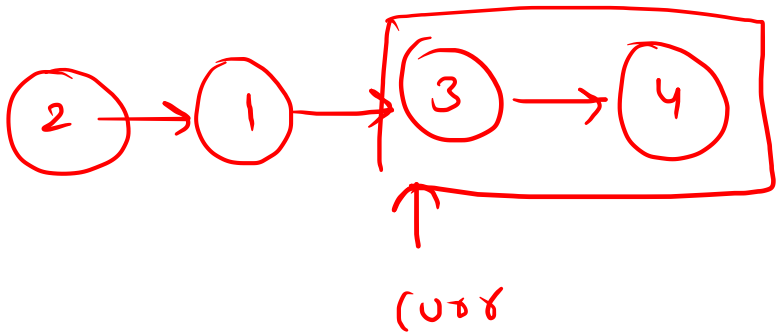


o/p :

```
int temp_data = curr -> next -> data;
curr -> next -> data = curr -> data
curr -> data = temp_data
```

Run →

[curr = curr → next → next



Costly

NOT Efficient

(Swapping data)

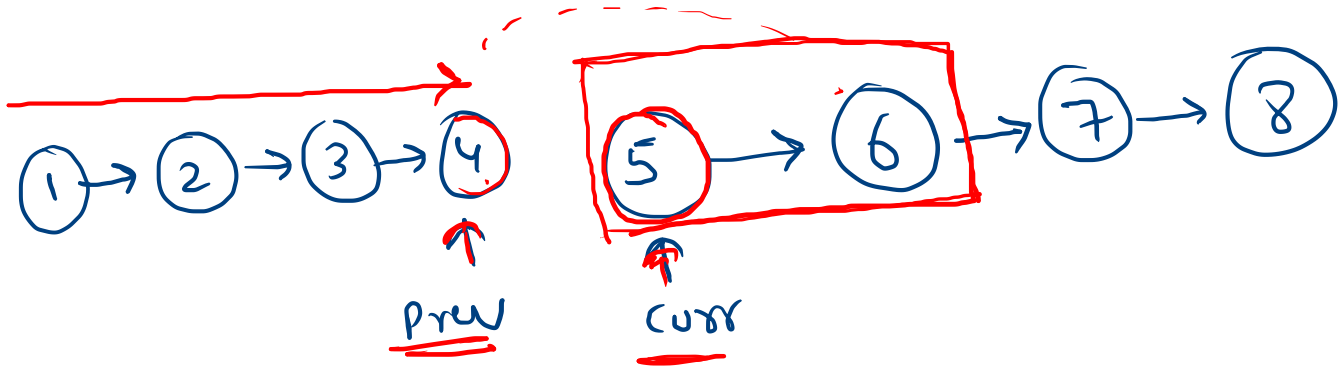
-> It may costly task if data is too large

Best method

I/p : 1 → 2 → 3 → 4 ✓

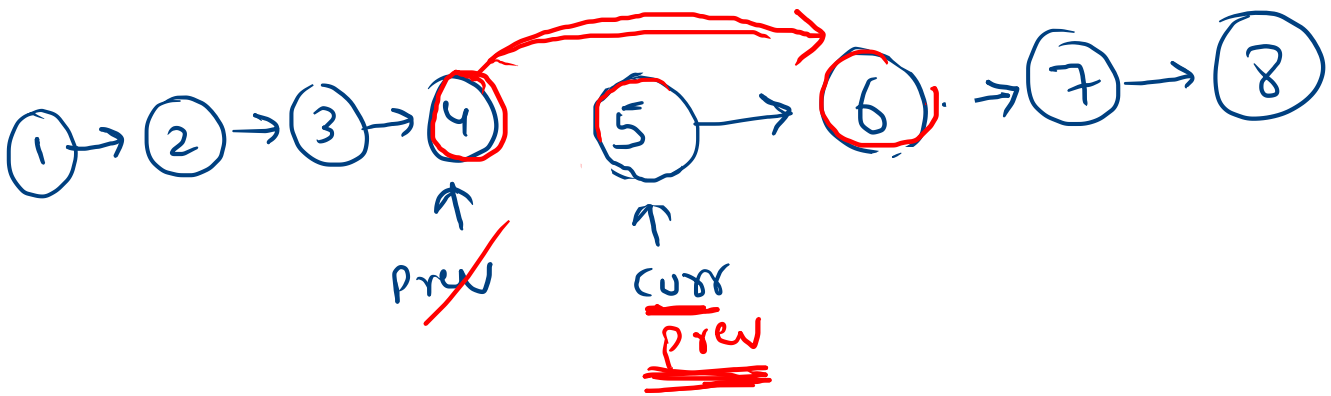
Seq^n.

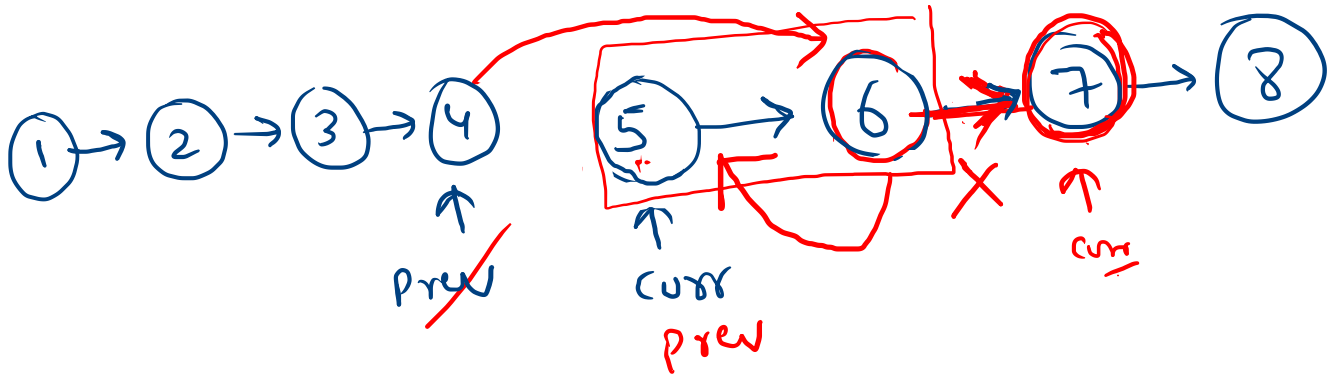
desn

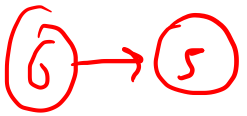


⇒ prev → next = curr → next

prev = curr

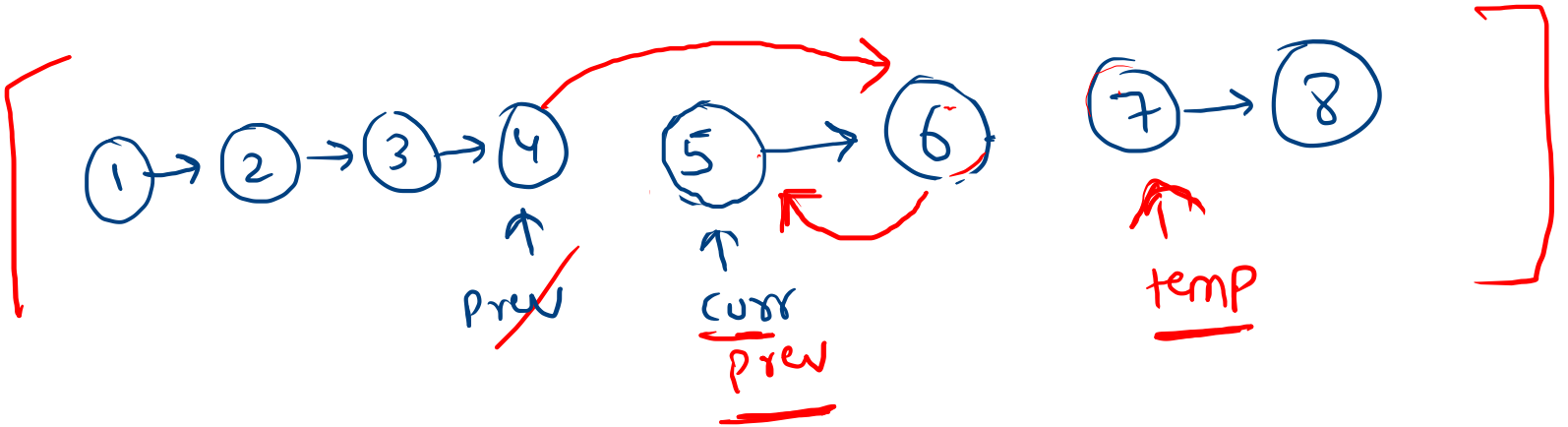


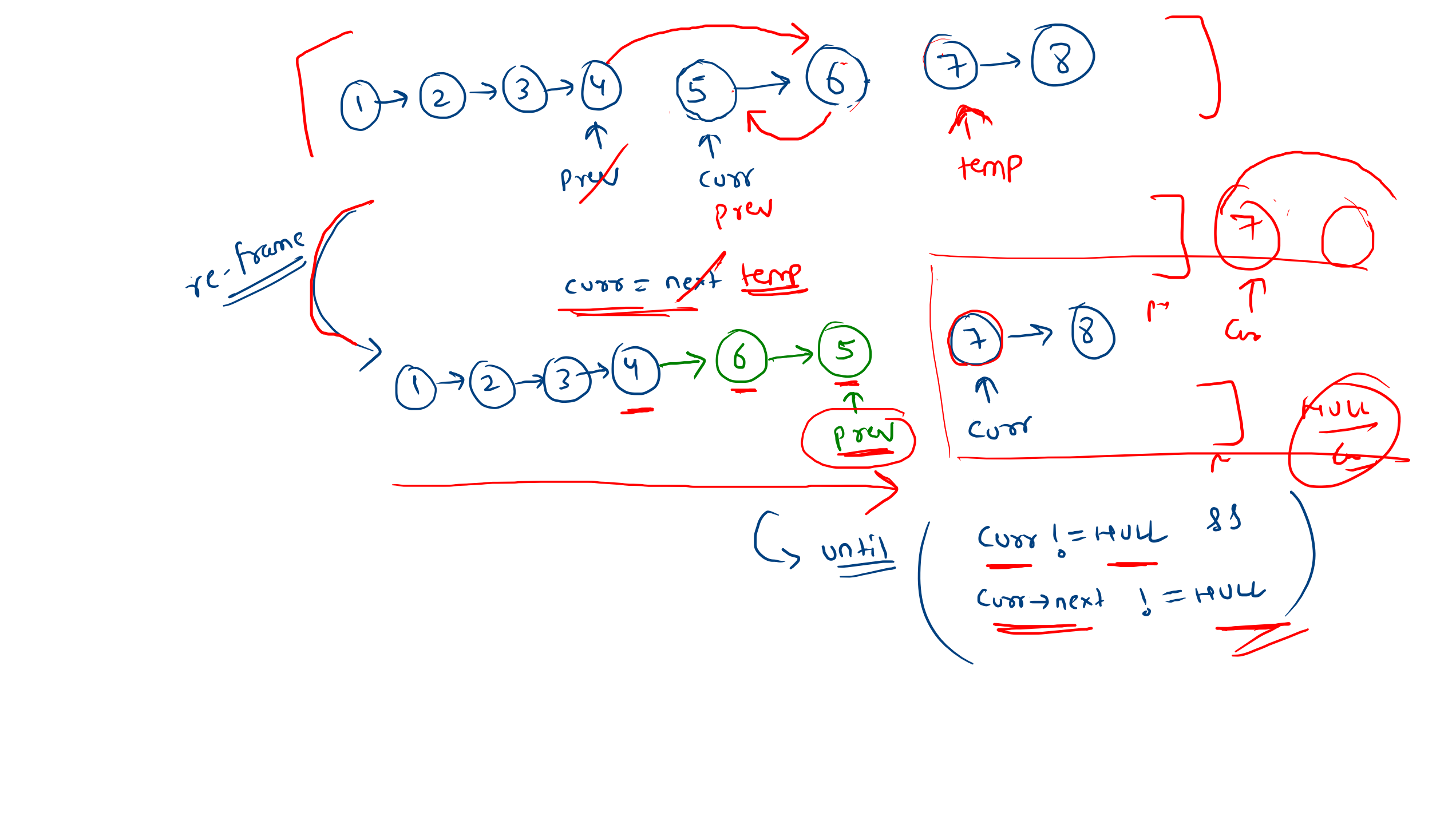




Node* temp = curr → next → next;

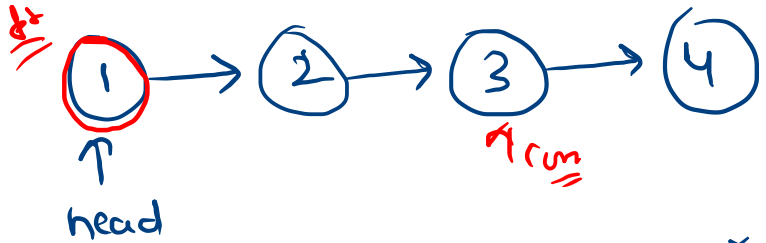
curr → next → next = curr




1 2 3 4 5 6 7 8
prev
curr
temp
re-frame
curr = next
temp
1 2 3 4 6 5
prev
curr
NULL
until
curr != NULL
curr->next != NULL

Edge case:

Ilb:

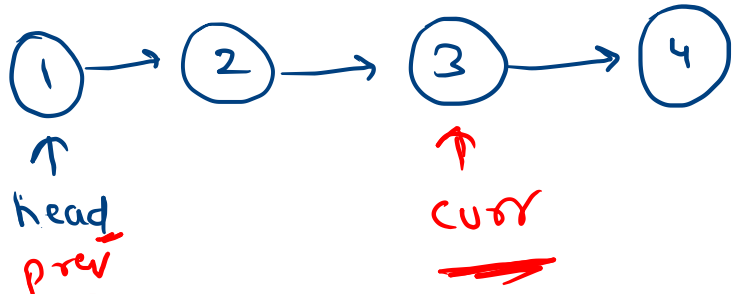


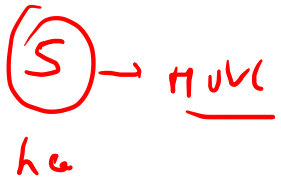
(head
head→next)

⇒ Node* curr = head→next→next

Node* prev = head

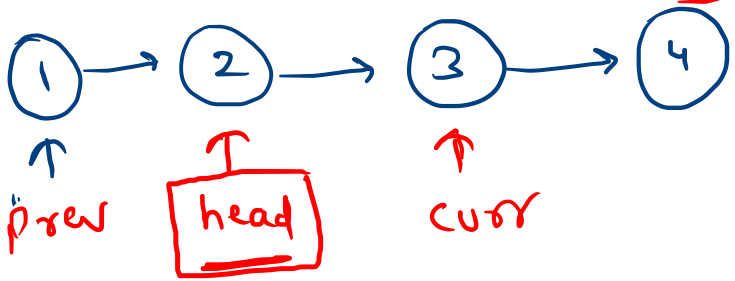
⇒

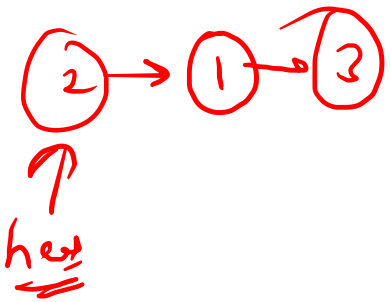


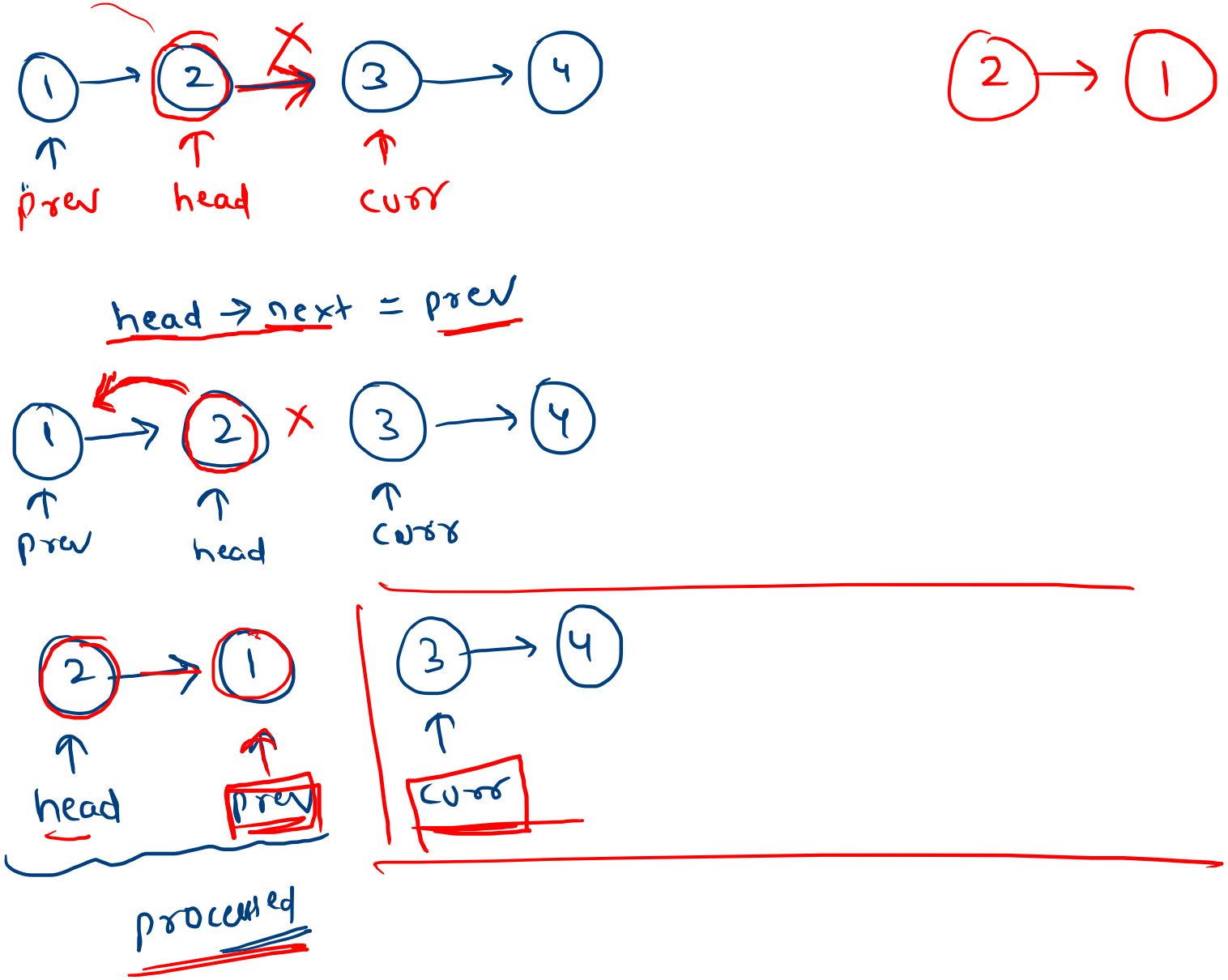


⇒ Ultimately, head should be 2nd node

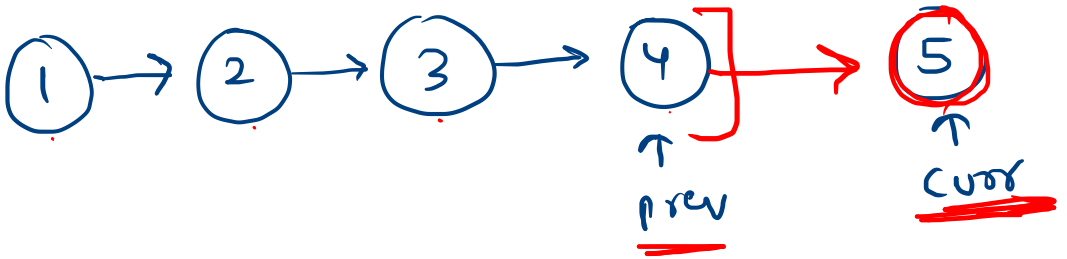
head = head→next



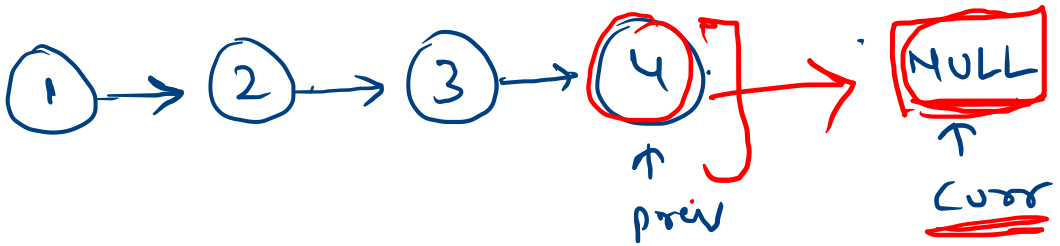



1
2
3
4
prev
head
curr
2
1
head → next = prev
1
2
3
4
prev
head
curr
diagram
2
1
head
prev
3
4
curr
processed

Edge case



Odd



Even

proceed

```
{
    prev->next = curr
    return head;
}
```