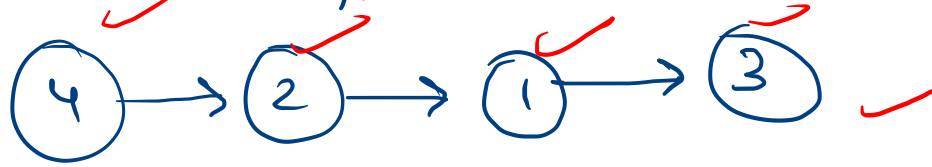
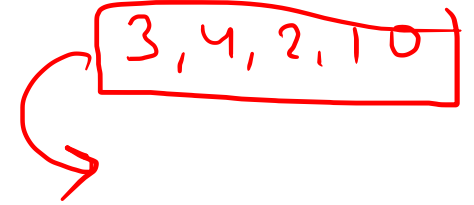
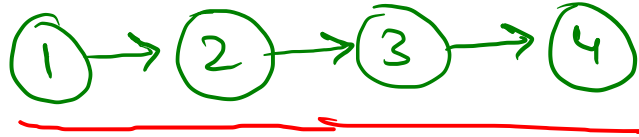


# Sort Linked List

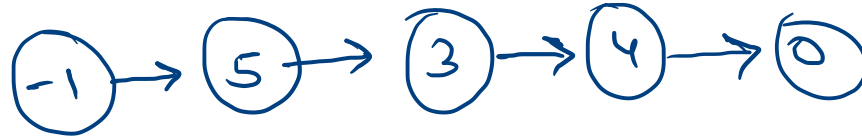
I/p:



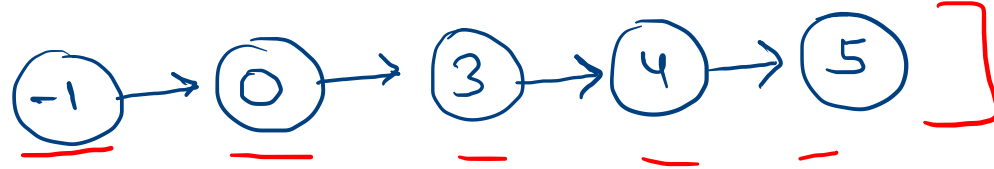
o/p:



I/p:



o/p:



Naive  
✓ NAIVE  
APPROACH

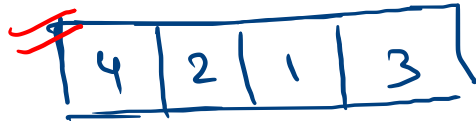
Qp:



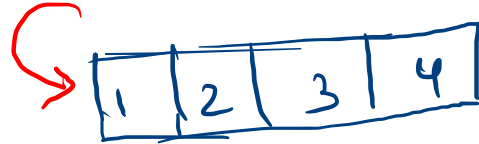
Sol<sup>n</sup>:

vector<int> data;

→ Traverse Linked list and Store node → val in data;



→ sort the vector ⇒ sort ( data.begin(), data.end() )



→ Traverse linked list and replace node → val = data[i]  
[ i → 0 to 3 ]

Naive  
NAIVE  
APPROACH

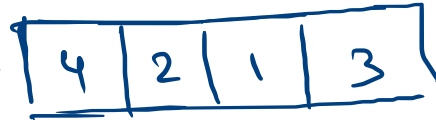


Sol<sup>n</sup>: Vector<int> data;

→ Traverse Linked list and Store node → val in data;

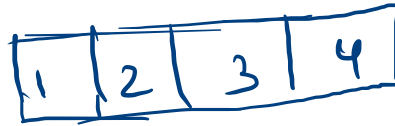
Space  
Complexity

$= O(n)$



—  $O(n)$  ✓

→ sort the vector  $\Rightarrow$  sort (data.begin(), data.end())



—  $O(n \log n)$  ✓

→ Traverse Linked list and replace node  $\rightarrow$  val = data[i]

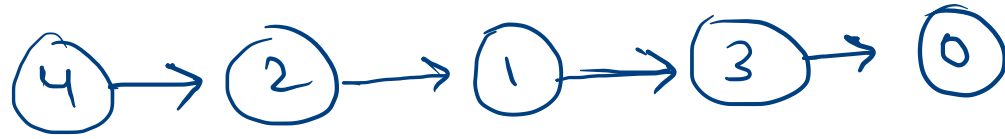
—  $O(n)$

$\Rightarrow$  Time  $\Rightarrow$   $O(n \log n)$

Optimised  
Sol<sup>n</sup>

⇒

Ex:

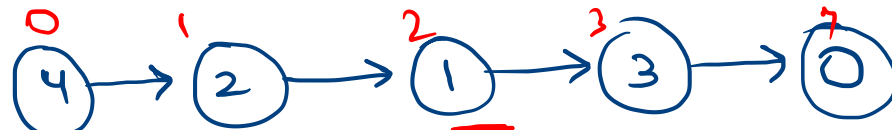


we will implement Merge Sort

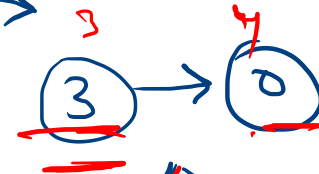
⇒ Merge Sort ⇒ We divide the Linked List in  
each single node

⇒ Now we merge the divided single nodes  
in sorted manner.

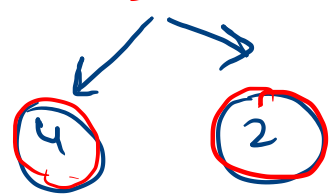
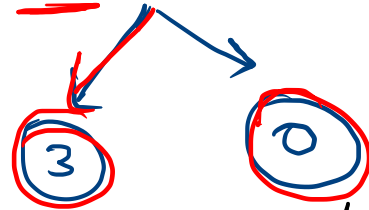
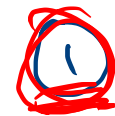
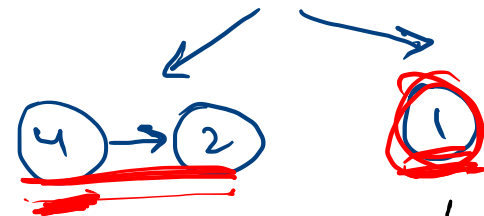
ILP



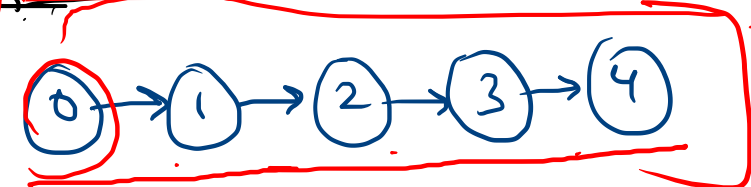
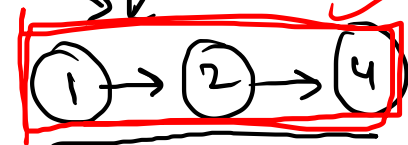
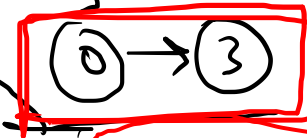
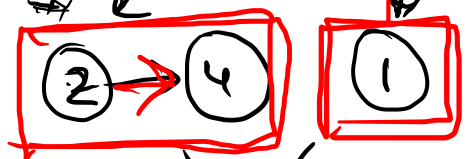
Divide phase



$(0+4)/2 = 2$   
 $0-2 \mid \underline{3-4}$

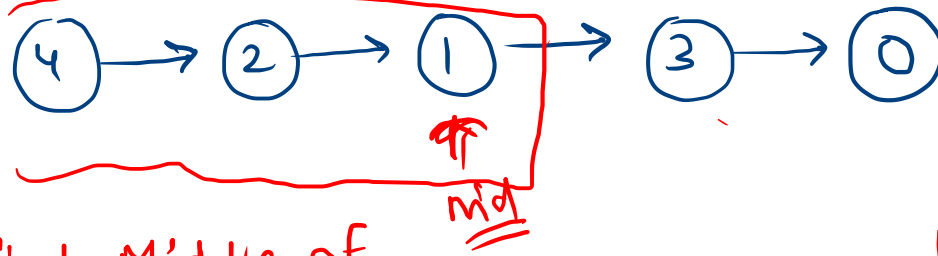


Conquer phase

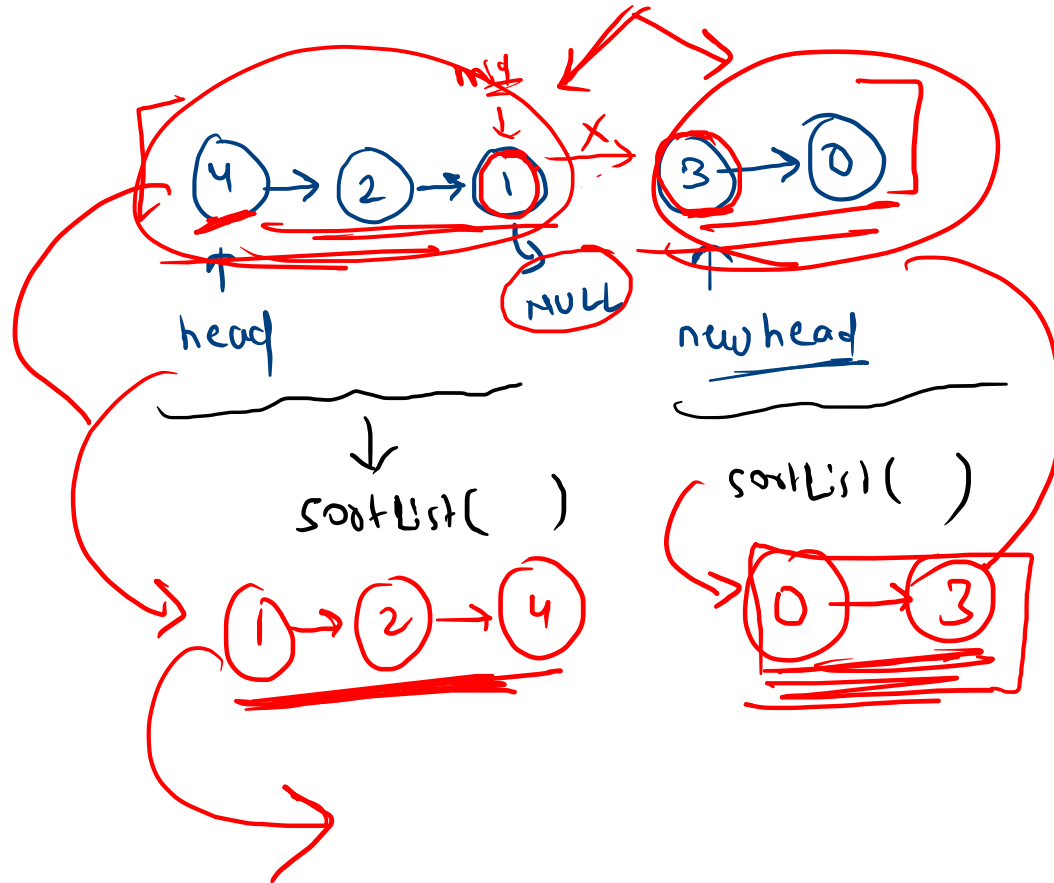


Sorted Linked List

Imp:



Step 1: Find Middle of  
Unked List



void SortList( Node\* head)

↳ Node\* mid = Mid( head )

Node\* newhead = mid → next;

mid → next = NULL

Node\* left-half = SortList( head );

Node\* right-half = SortList( newhead );

return merge sorted lists ( left-half,  
right-half );

```

Node* mergeSortedList( Node* P1, Node* P2)
{
    → if ( P1 == NULL or P2 == NULL )
        return (P1 == NULL) ? P2 : P1;
}

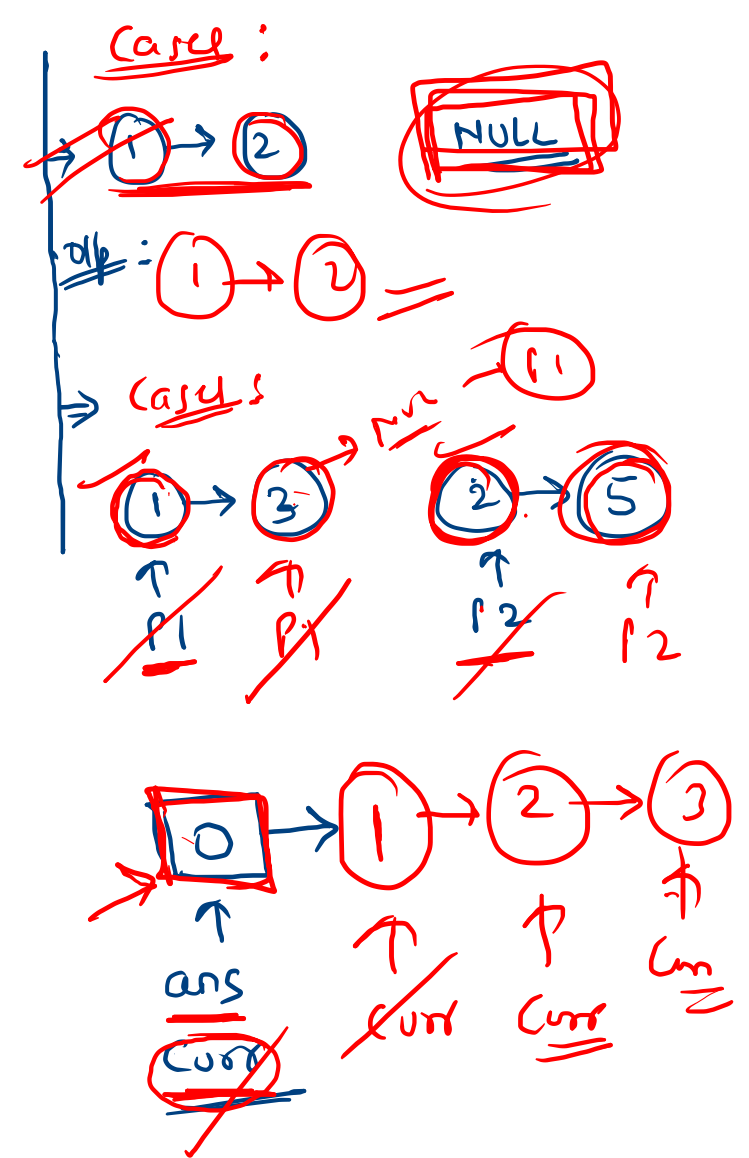
```

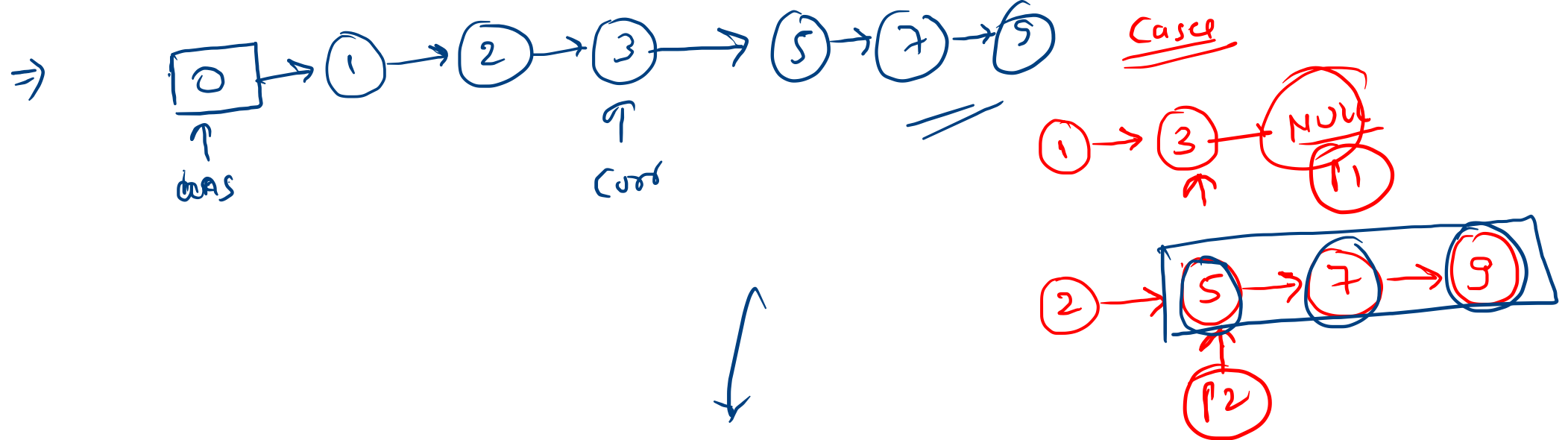
Wh!ker

```

is → if ( P1->val < P2->val )
{
    curr->next = P1;
    P1 = P1->next;
}
else {
    curr->next = P2;
    P2 = P2->next;
}
→ curr = curr->next;
→

```





⇒

$$\left[ \begin{array}{l} \text{if ( } \boxed{P1 \neq \text{NULL}} \text{ or } P2 \neq \text{NULL} ) \\ \boxed{\text{cur} \rightarrow \text{next}} \equiv \underline{(P1 \neq \text{NULL}) ? (P1) : (P2)} \end{array} \right]$$