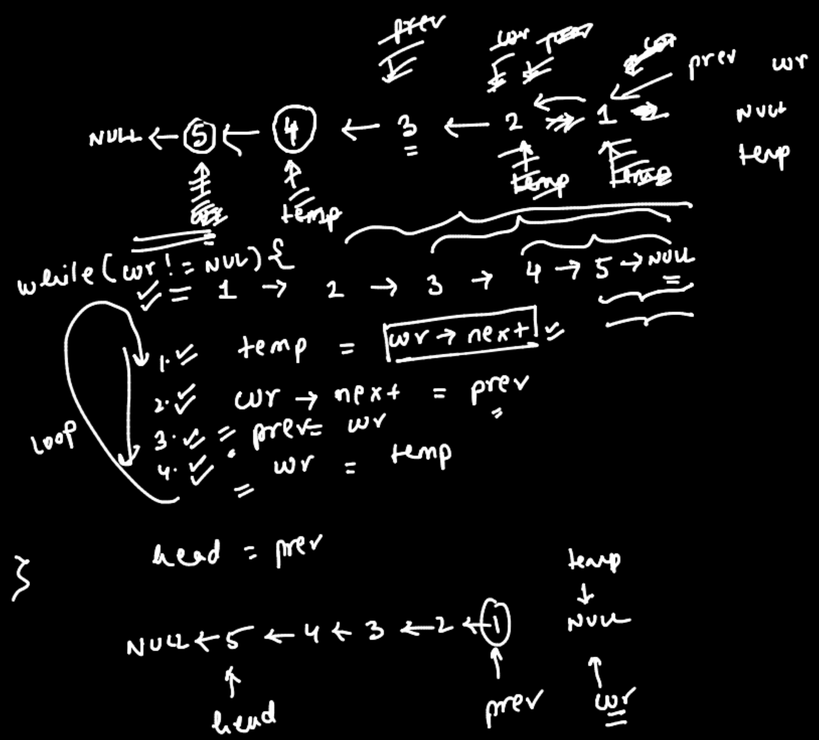
(1.) Reverse LL

I/P: 5 → 4 → 3 → 2 → 1

↑ head

O/P: 1 → 2 → 3 → 4 → 5

↑ head



2. Sorted Linked List

head1

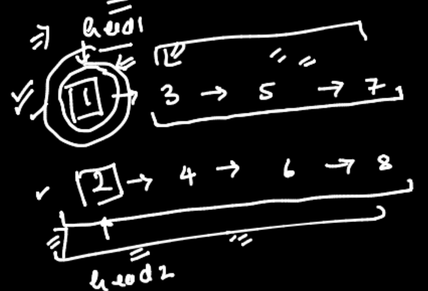
1 → 3 → 5 → 7

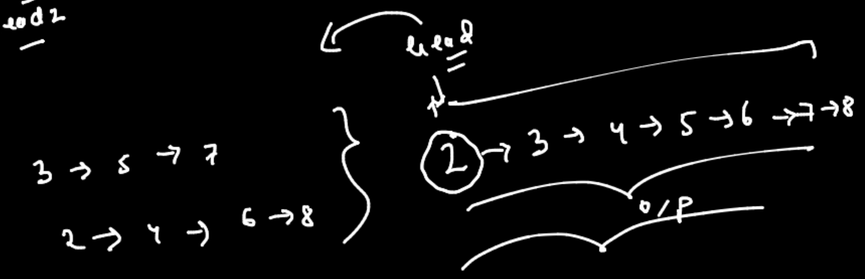
2 → 4 → 6 → 8

↑ head2

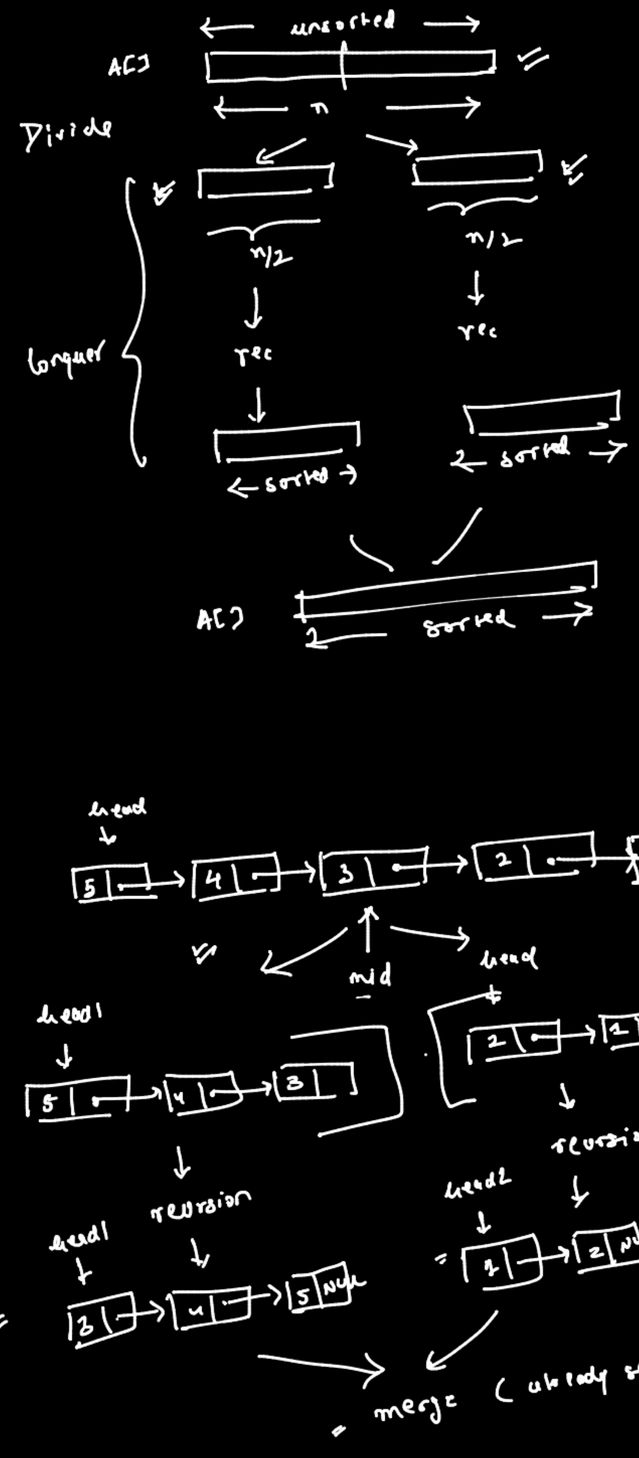
O/P: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8

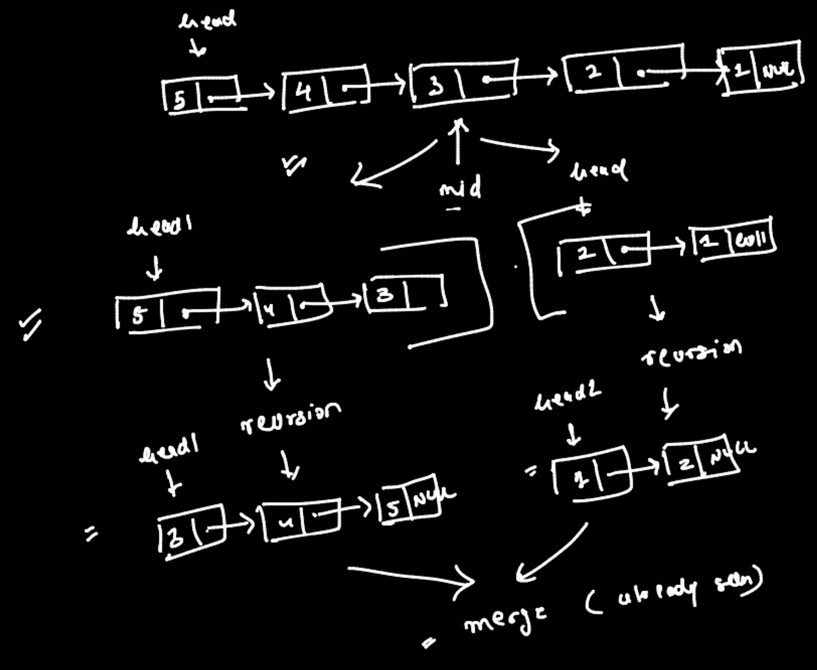
↑ head

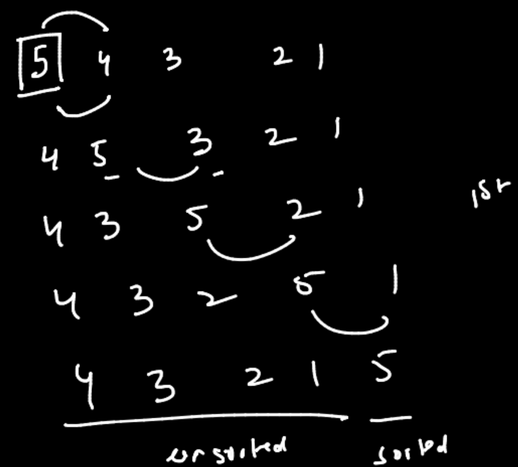


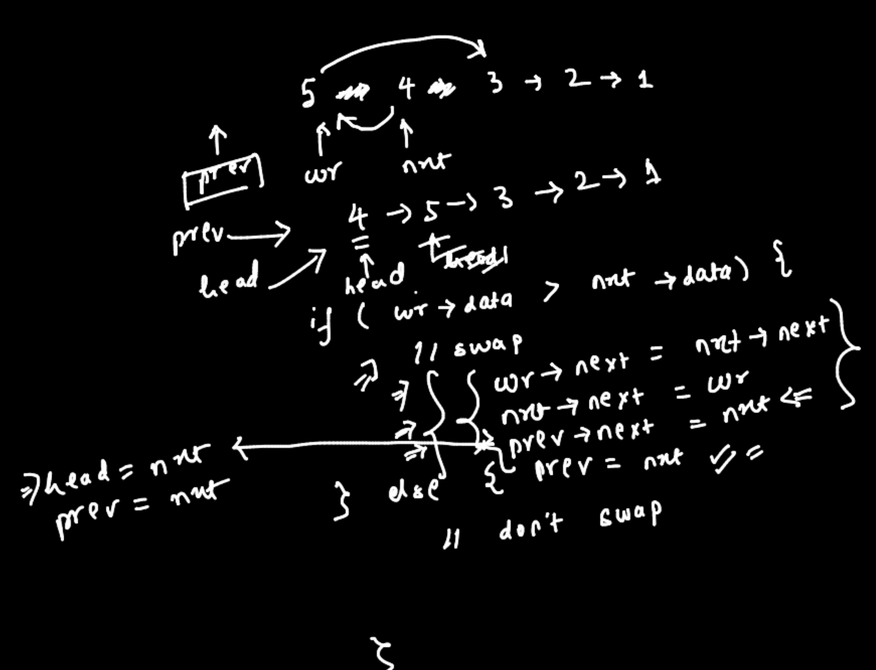


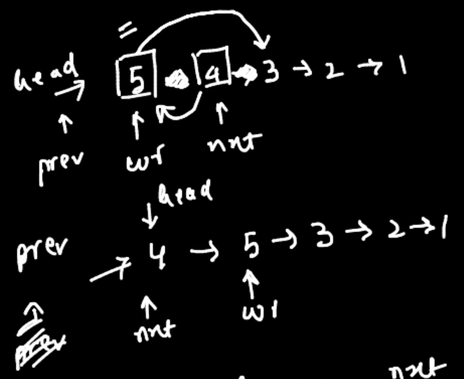
3. MergeSort

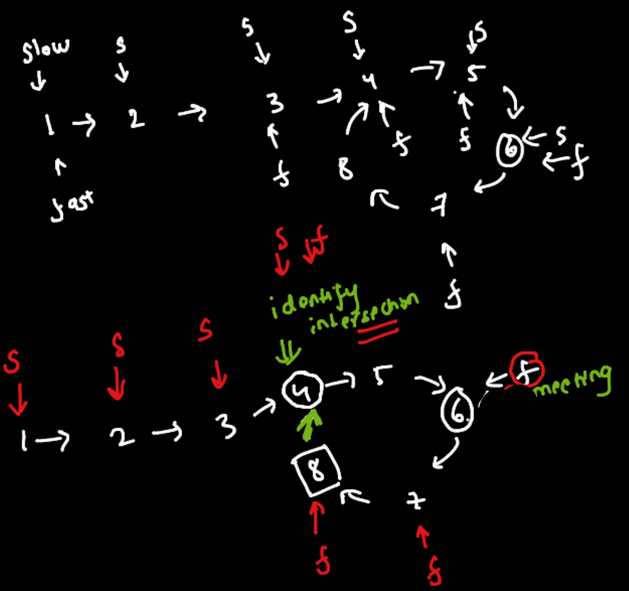

A[]
unsorted
n
Divide
n/2
n/2
rec
rec
Longer
sorted
sorted
A[]
sorted

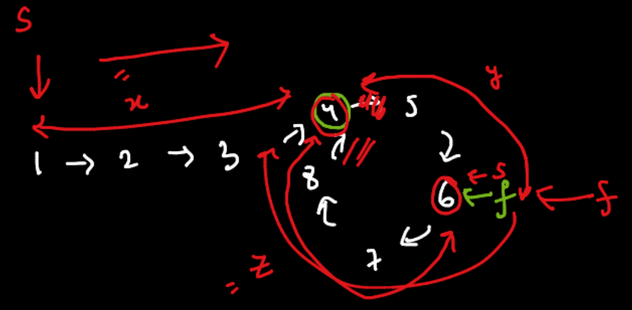

head
mid
head
head1
recursion
head1
recursion
head2
recursion
merge (already sorted)


5 4 3 2 1
4 5 3 2 1
4 3 5 2 1
4 3 2 5 1
4 3 2 1 5
unsorted
sorted
ist


5 → 4 → 3 → 2 → 1
prev
cur
nxt
prev
head
4 → 5 → 3 → 2 → 1
head
head1
if (cur → data > nxt → data) {
// swap
cur → next = nxt → next
nxt → next = cur
prev → next = nxt
prev = nxt
head = nxt
prev = nxt
} else {
// don't swap
}

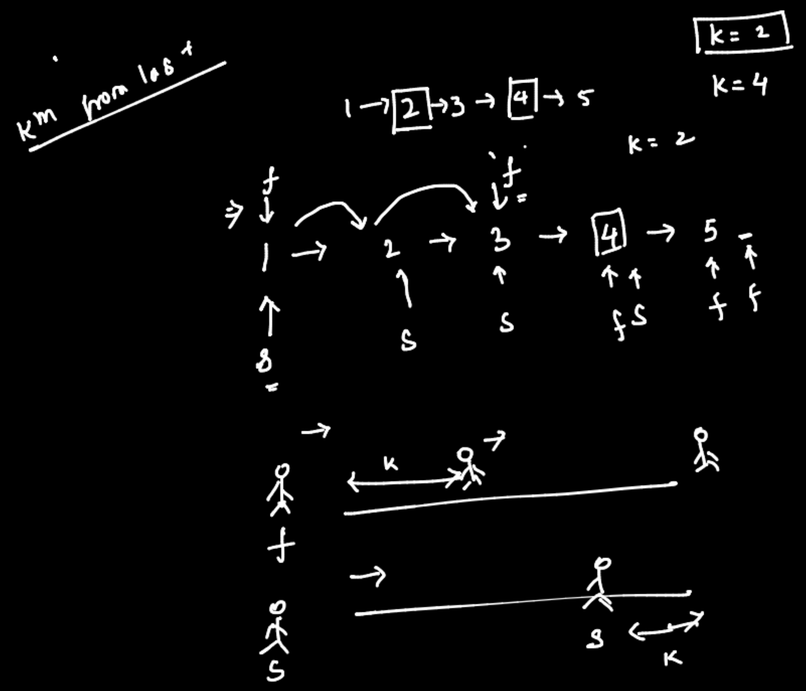






$$\Rightarrow \nabla_s = x+y = \Rightarrow \nabla_f = x+2y+z =$$

$$\Rightarrow S_f = 2S_s \Rightarrow \Rightarrow \nabla_f = 2\nabla_s$$

$$x+2y+z = 2(x+y) \Rightarrow \cancel{x+2y} + z = \cancel{2x+2y}$$

$$(x = z)$$