



## 1721. Swapping Nodes in a Linked List

Medium

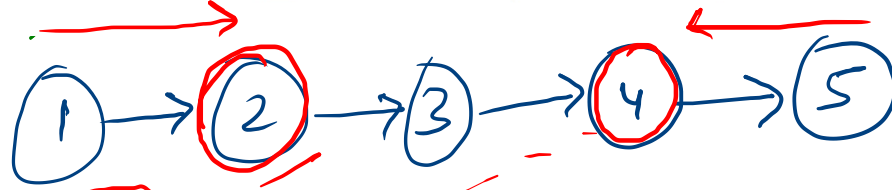
3007

110

Add to List

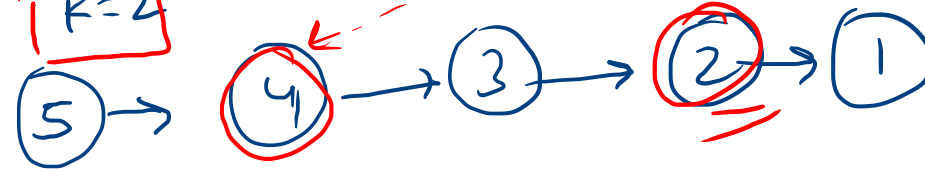
Share

I/p:



k=2

O/p:



$\neq \left[ \begin{array}{l} k^{\text{th}} \text{ from front} \\ k^{\text{th}} \text{ from End} \end{array} \right]$

I/p:

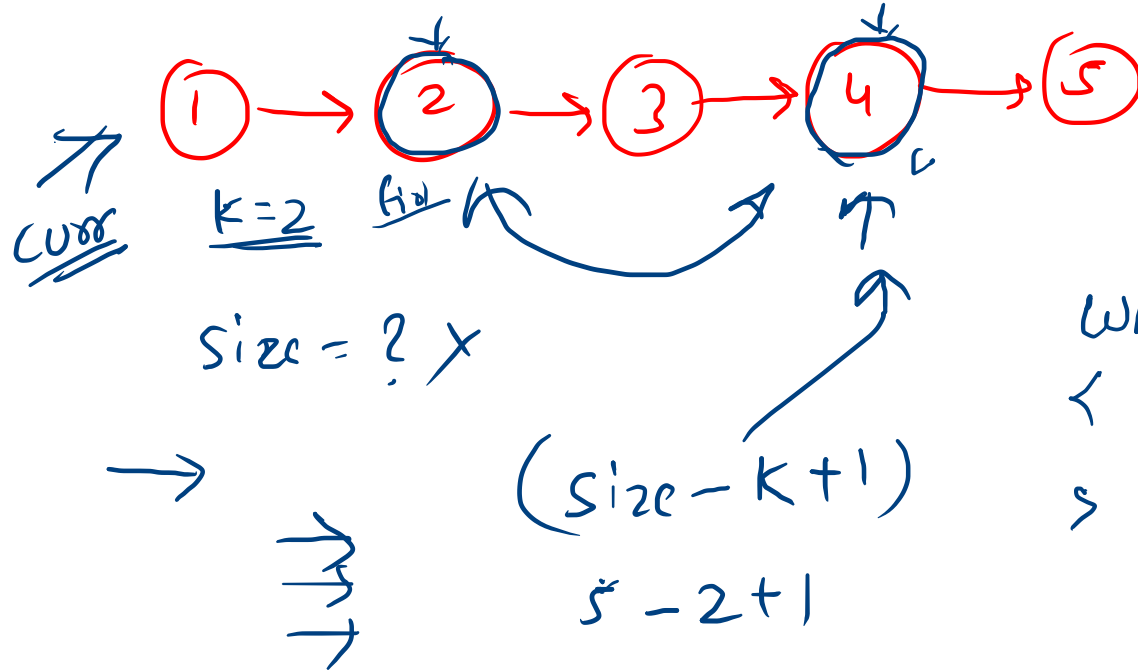
[7, 9, 6, 6, 7, 8, 3, 0, 9, 5]

k=5

O/p:

7 9 6 6 8 7 3 0 9 5  
↑  
here

Give  
So



JO

```
while (curr != NULL)
{
    count++;
}
```

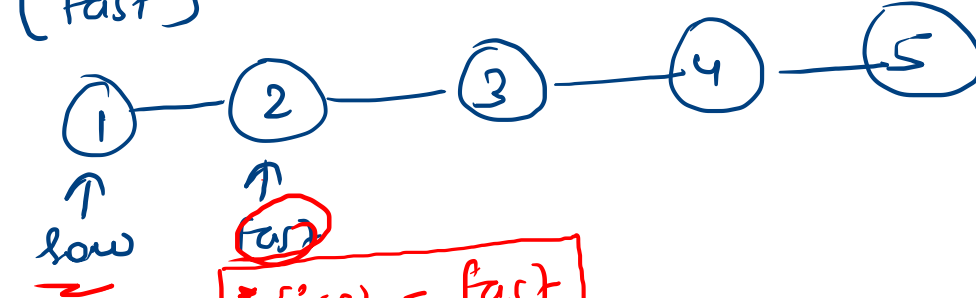
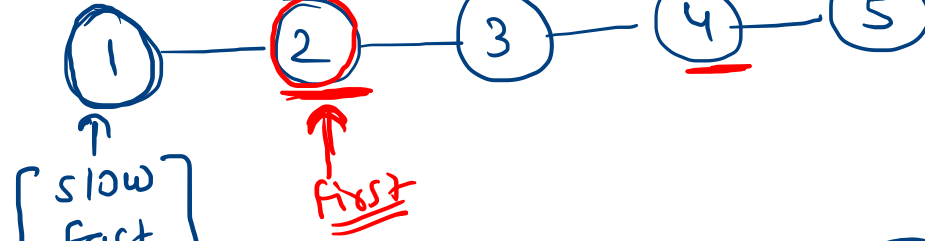
Swap (first  $\rightarrow$  data, second  $\rightarrow$  data)

Sol<sup>n</sup>:

→ Last lecture → "delete n<sup>th</sup> Node from End"

Q/p:

Sol<sup>n</sup>:



Imp<sup>ts</sup>

⇒

\* first = fast

K=2

fast  
K=2

```
while(--K)  
< fast = fast → next  
>
```

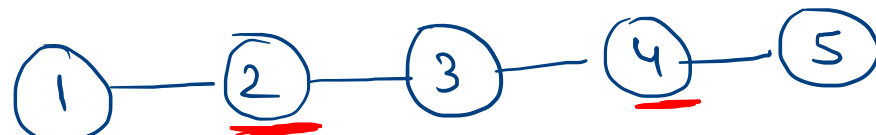
K=2

Sol<sup>n</sup>:

→ Last lecture → "delete  $n^{\text{th}}$  Node from End"

K=2

inp:

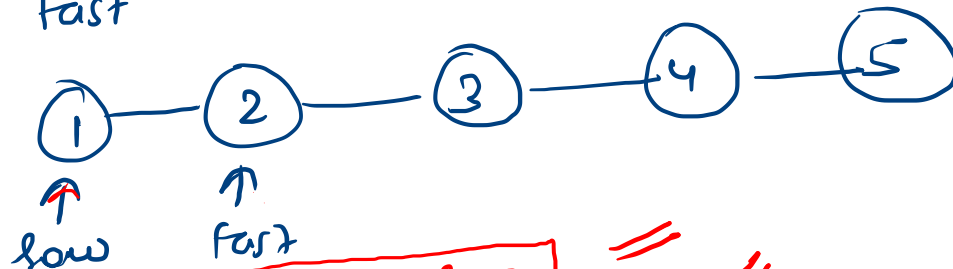


Sol<sup>n</sup>:

slow  
fast

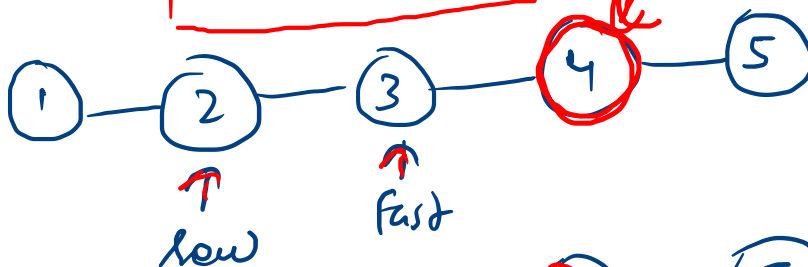
Imp →

⇒

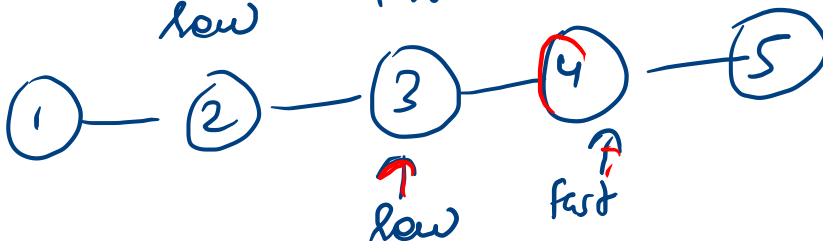


\* first = fast

⇒



⇒

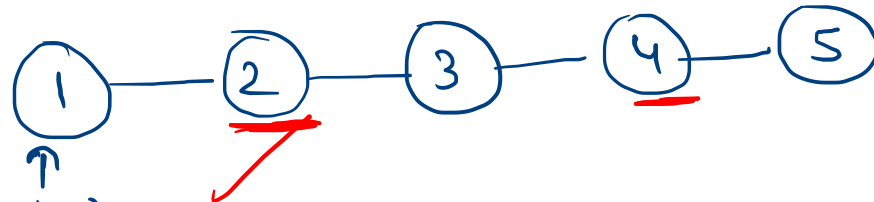


```
while(--K)
< fast = fast->next
>
```

Sol<sup>n</sup>:

→ Last lecture → "delete n<sup>th</sup> Node from End"

Q/p:

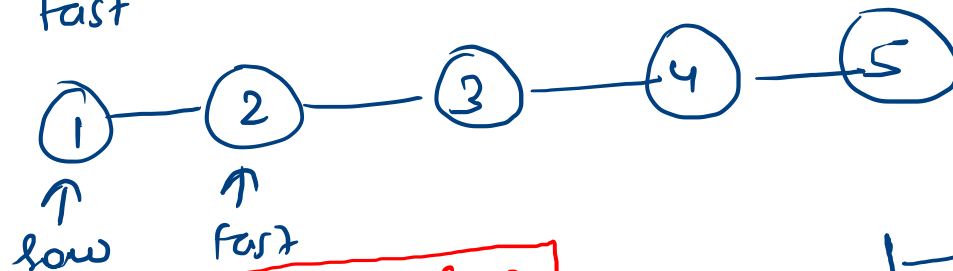


Sol<sup>n</sup>:

slow  
fast

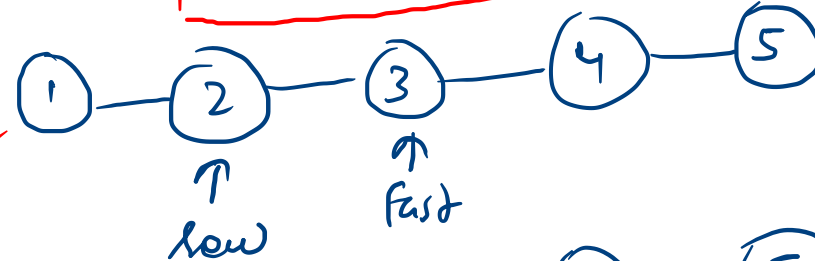
Imp →

⇒

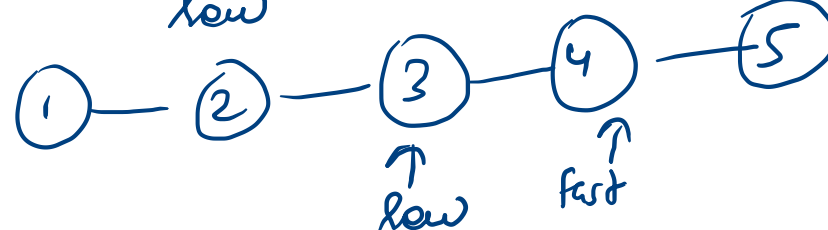


\* first = fast

head ⇒



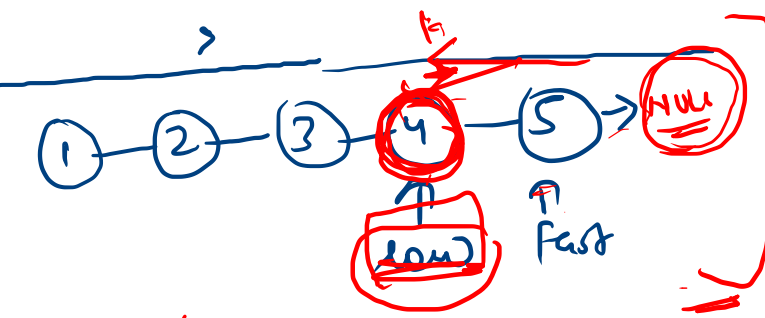
⇒



K=2

while (fast → next  
!= null)  
{

while (--K)  
< fast = fast → next



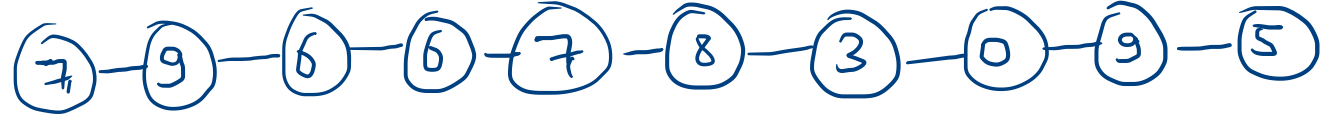
swap ( first → val,  
slow → val )

return head

Home  
work

Day  
Run

IP:



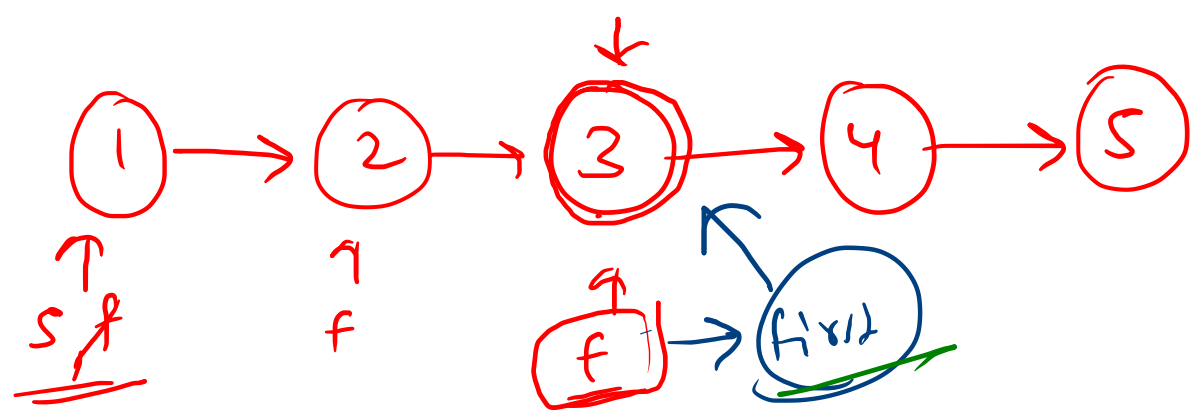
K=5

⇒ 

* <u>slow</u>
* <u>fast</u>
* <u>first</u>

 ⇒

$k=3$   
 while (true)  
 { fast = fast + next  
 }



first = fast

