

19. Remove Nth Node From End of List

Medium

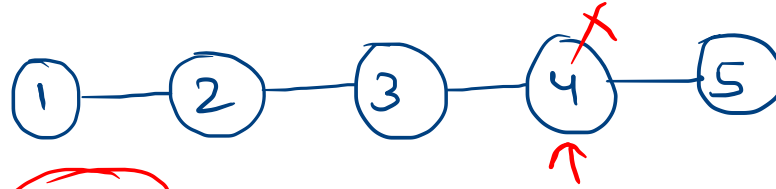
11388

519

Add to List

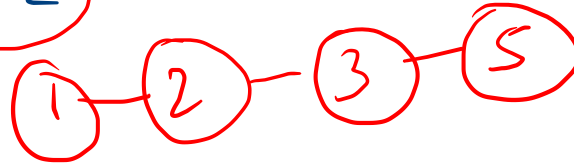
Share

Input:

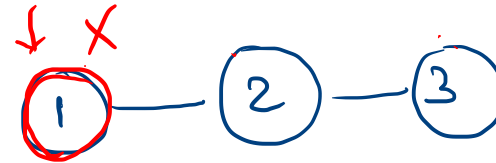


n=2

Output:

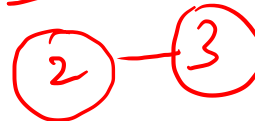


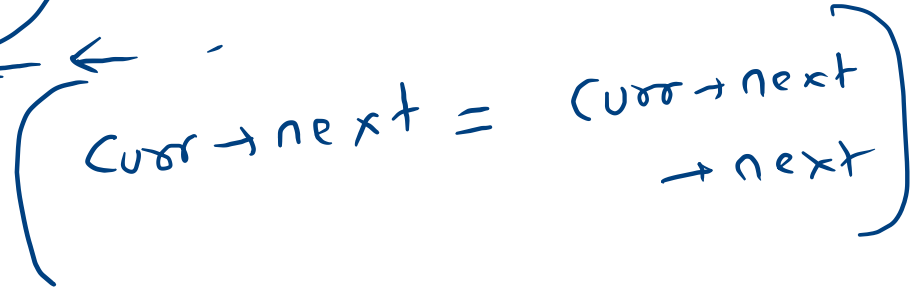
Input:



n=3

Output:

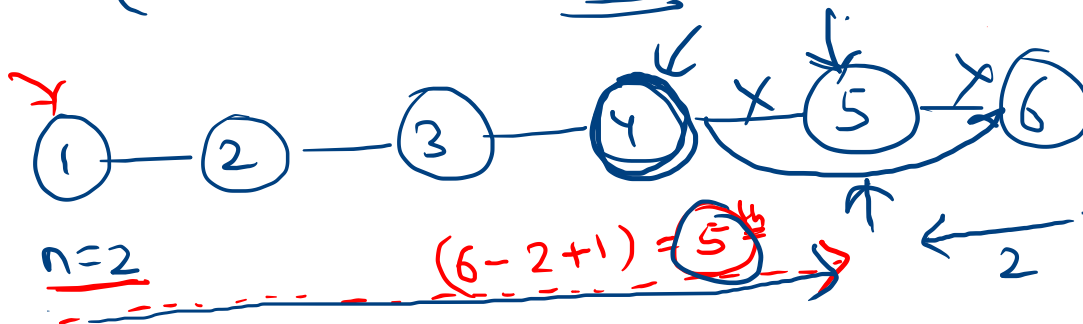




I^{sd} Approach

- > Count No. of Nodes (say, size)
- > delete (size - n + 1) Node

ILP:



o/p :

size = 6

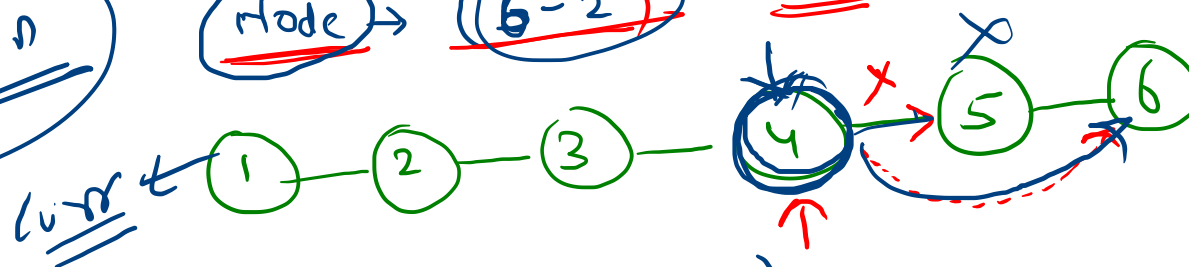
$$D = 2$$

mode

 ~~$(6-2)$~~

Node

desired-rod



while (desired-node == curr)

{ curr = curr → next

curr \rightarrow next

$n = 6$

```
while (count != 4000)
{
    count++;
}
```

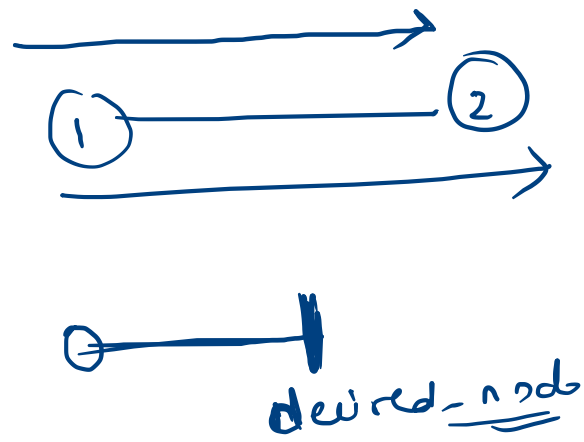


5(20 - n + 1)

$$6 - 2 + 1$$

5

Better

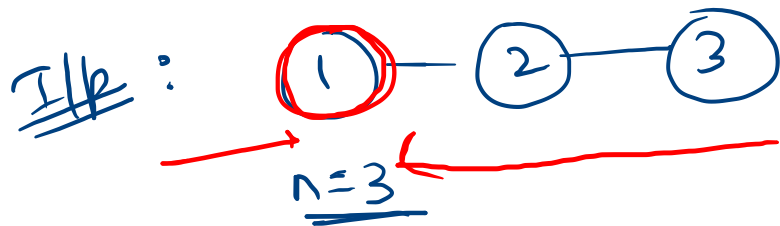


$O(N)$

$O(H)$

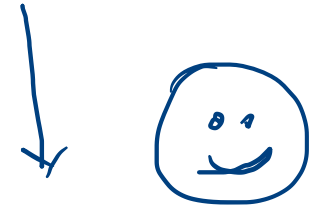


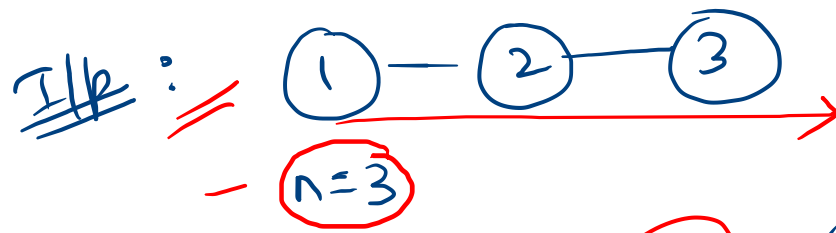
$O(H)$



\Rightarrow delete Node pos = $\frac{\text{size} - n + 1}{(3 - 3 + 1)} =$ 1st pos

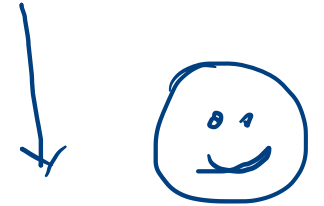
✓



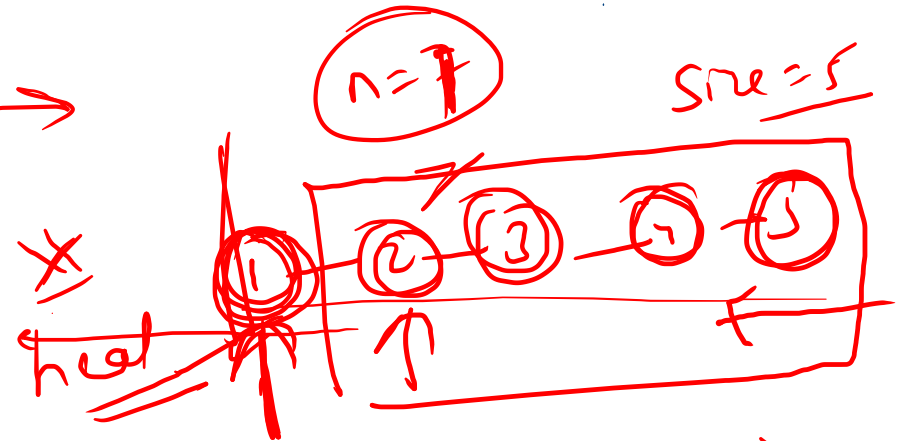


Size = 3

\Rightarrow delete Node $pos = (\text{Size} - n + 1)$
 $= (3 - 3 + 1) = 1^{st} \text{ pos}$



if ($\text{Size} == n$)
 return $\text{head} \rightarrow \text{next}$

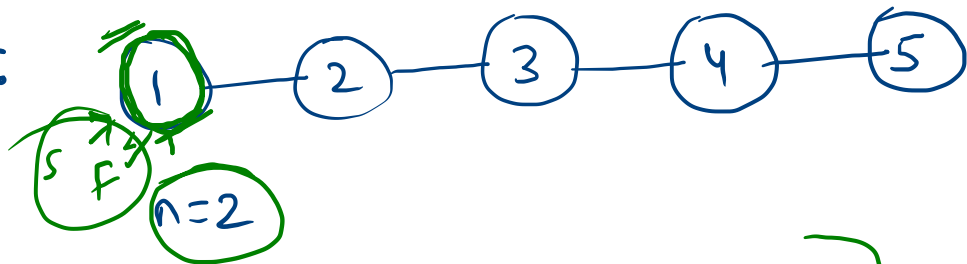


$(\text{Size} == n)$
 \uparrow \uparrow
Cal Size

2nd Approach

$s \rightarrow \text{slow}$
 $f \rightarrow \text{fast}$

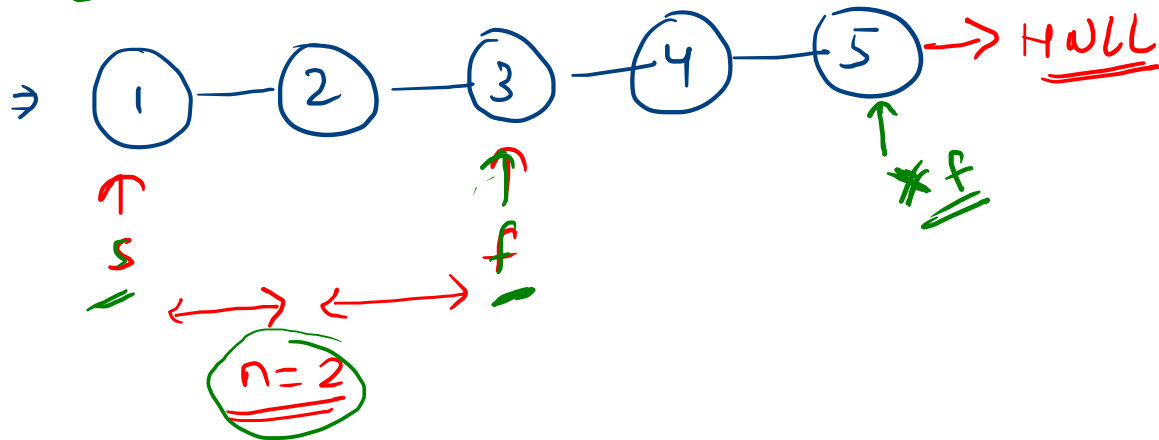
I/p:



$s \quad f$ 4

O/p:

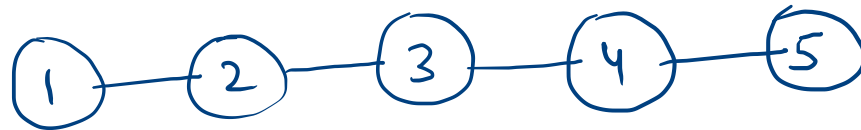
Two pointer Approach



2nd Approach

s → slow
f → fast

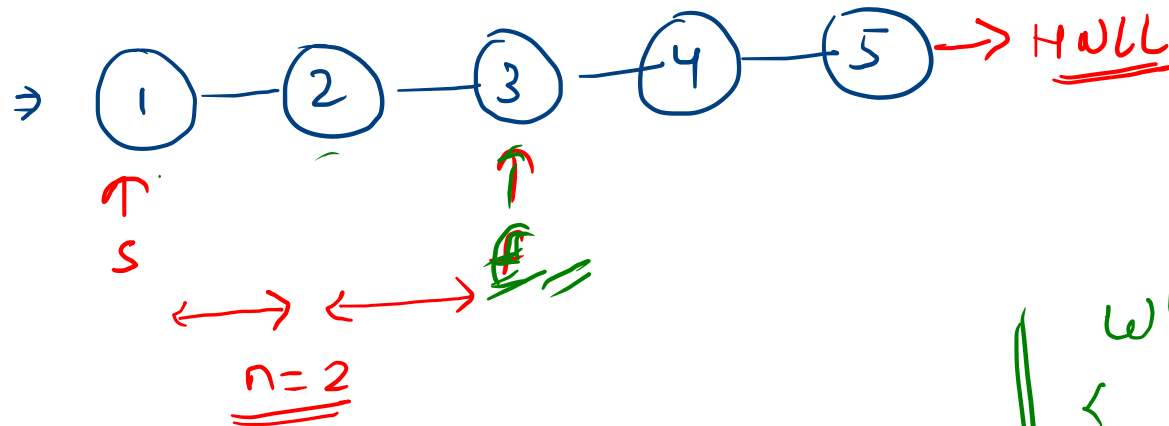
I/p:



n=2

O/p:

Two pointer Approach

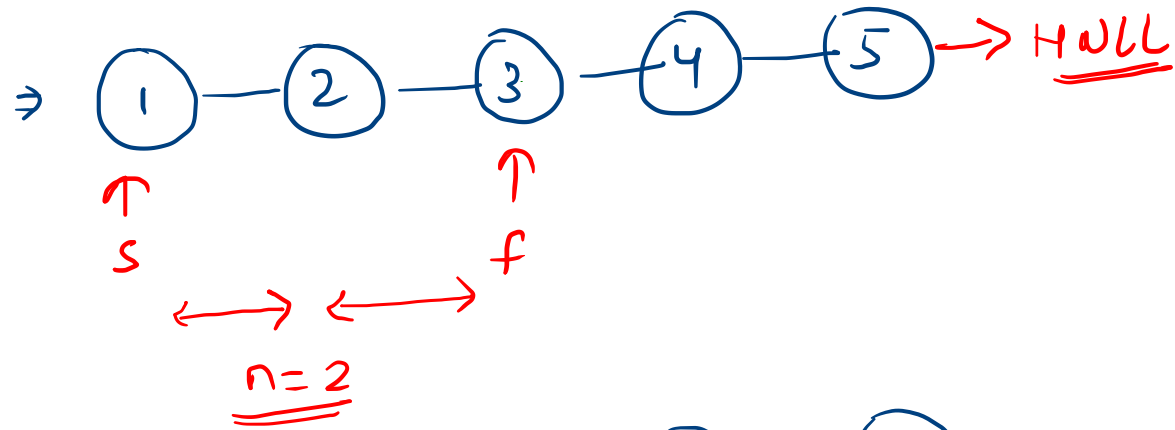


slow = head
fast = head

```
while( n-- ) {  
    fast = fast → next;  
}
```

(n=2)

Step 1:



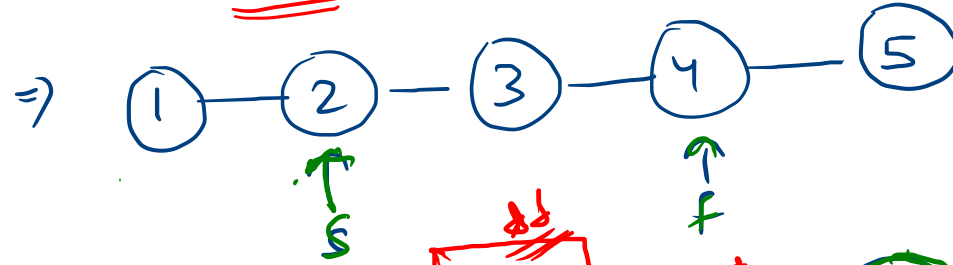
UnHl

fast → next ≠ HULL

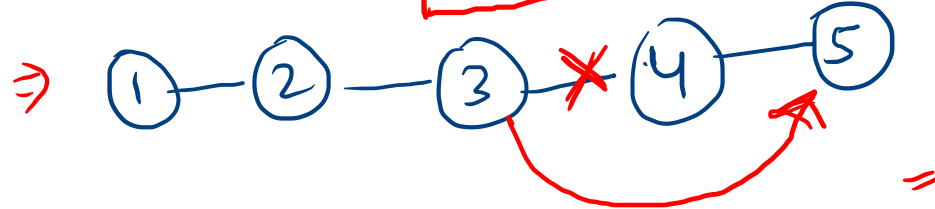
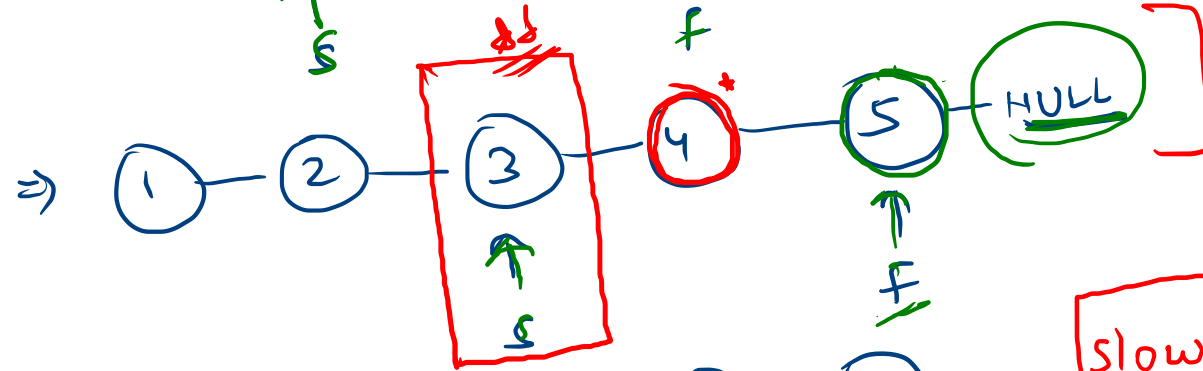
{ slow →
fast → } =

Single Speed

Step 2:



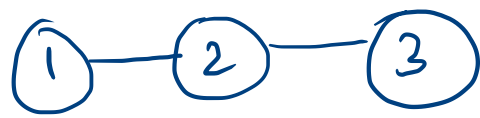
Step 3:



slow → next = (slow → next →
next)

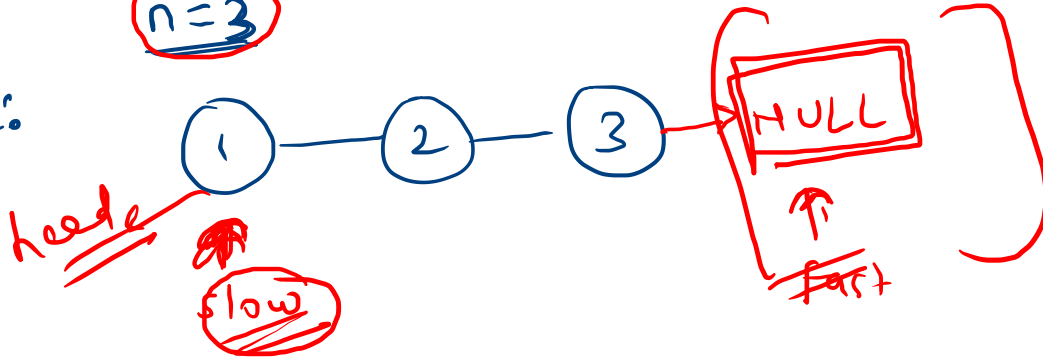
Ex:

Input:



n=3

Output:



if (fast == NULL)
 return slow → next Δ
 head → slow

A/q

Value of n is
in b/w size of
linked list

n=3

fast → move