LeetCode #19

↳ Day - 6

Removing n^{th} Node from the Last

2nd Node from the Last

Example


```
  (1) ----→ (2.) ----→ (3) ---→ (4) ---→ (5)
   5th        4th        3rd       2nd       1st
                       n = 2
```

⇓

```
  (1) ---→ (2) ---→ (3) ---→ (5)
```
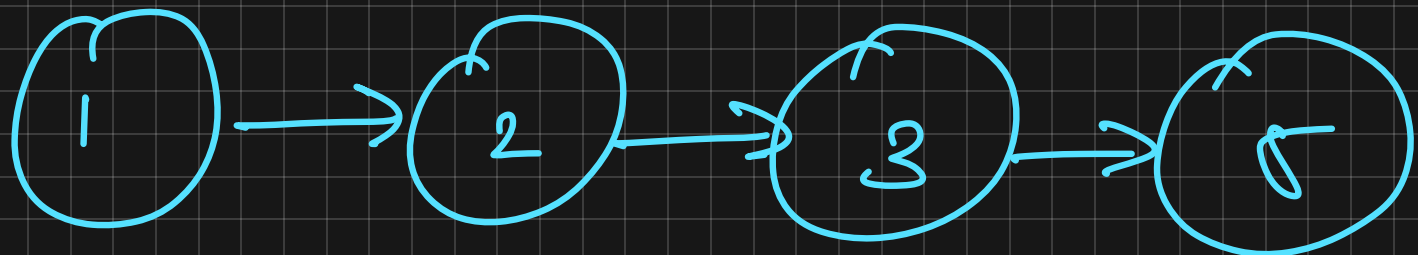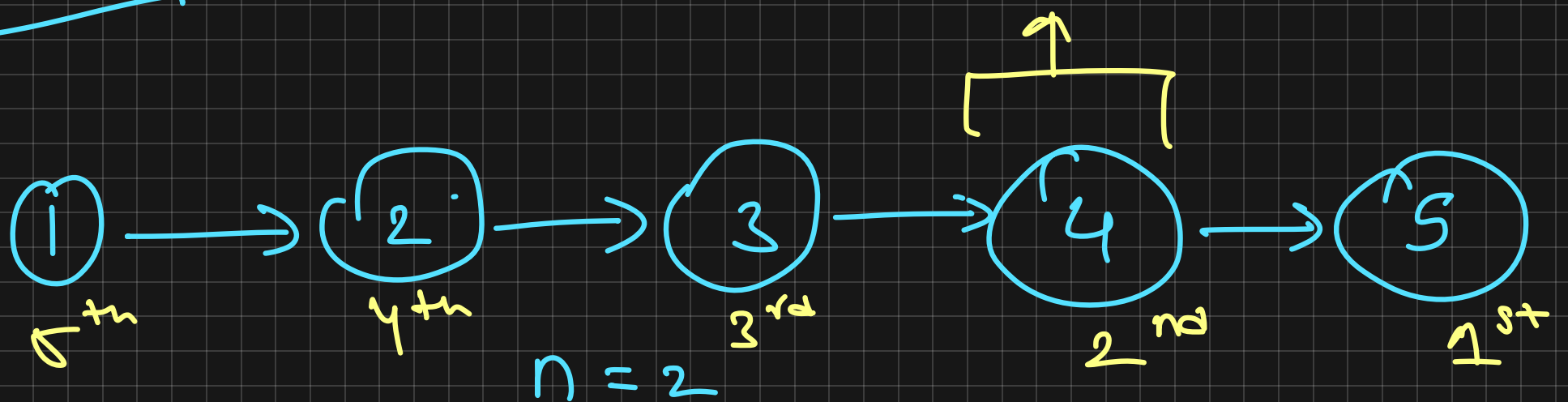
Approach

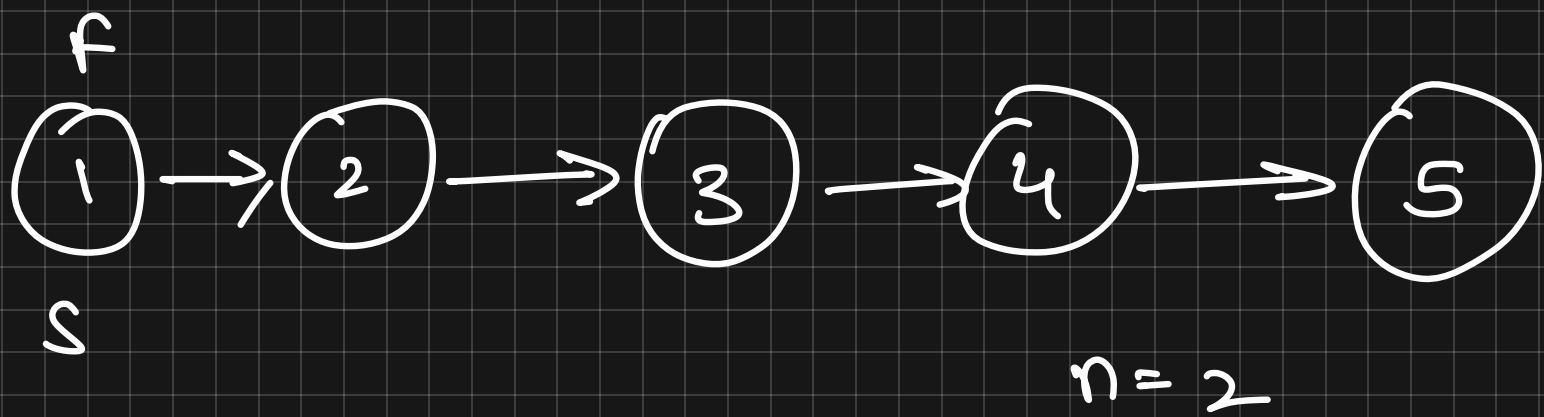① We Can use two pointer approach.

→ Fast Pointer

→ Slow Pointer

※ In this, (M-n) will always return the nth node

where M = length of LL.
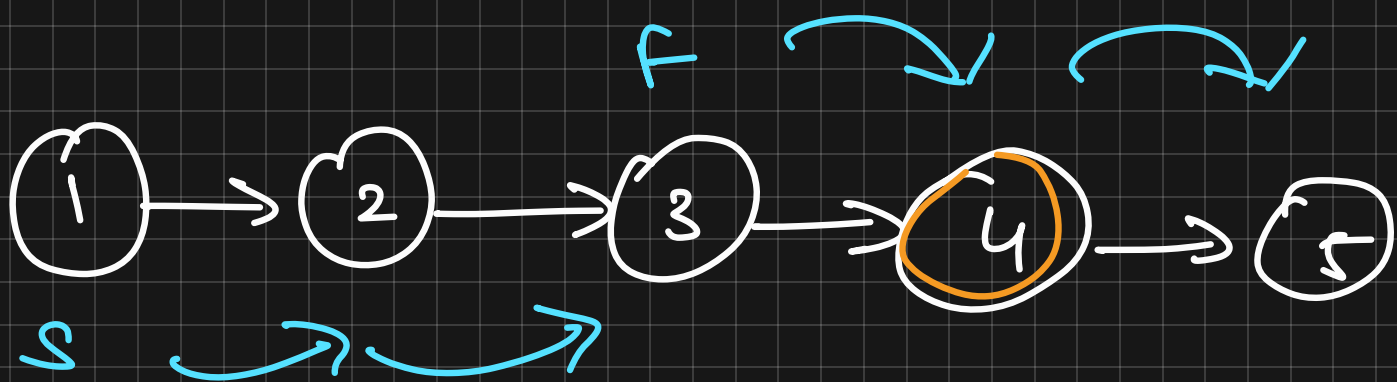
f

$(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5)$

s

$n = 2$

Head start for fast ptr

F

$(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5)$

s

Now   while   while (fast → next)

F

$(1) \rightarrow (2) \rightarrow (3) \rightarrow (4) \rightarrow (5)$

s
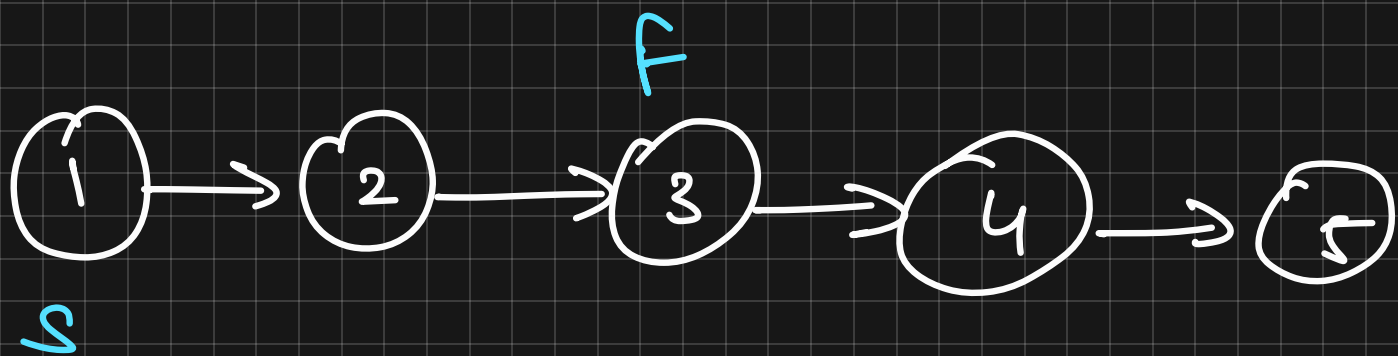
When f → next = NULL, it stops.

Now fast ptr is at the last Node.

and slow ptr is n^{th} step behind

fast ptr. So, the Node pointed by the

Slow ptr. will be (n^{th}) Node from

the end.

We remove it and then stitch back the remaining nodes.

①→②→③———→⑤ next → next

next → Removed

③ → next = ③ → next → next