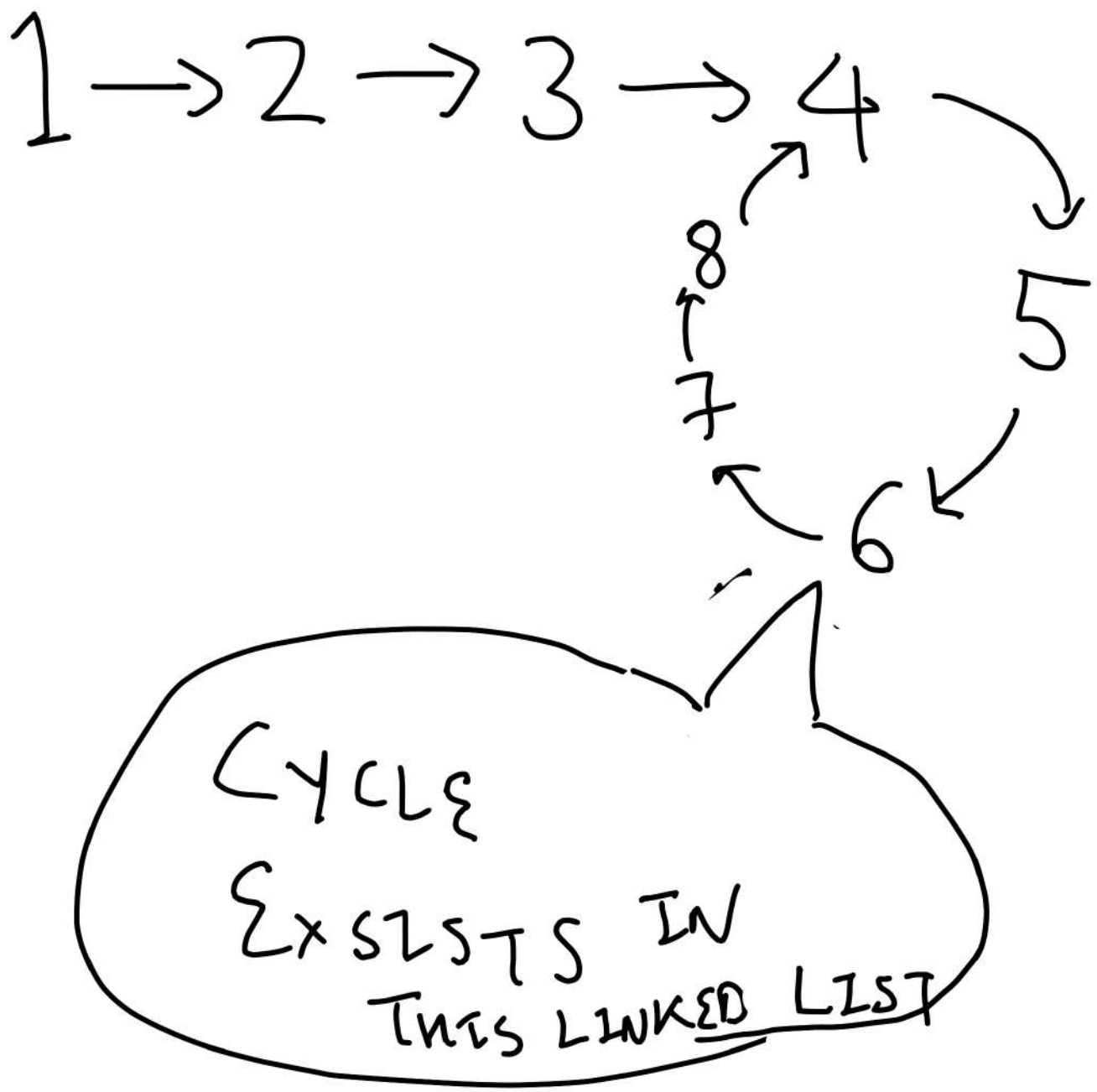
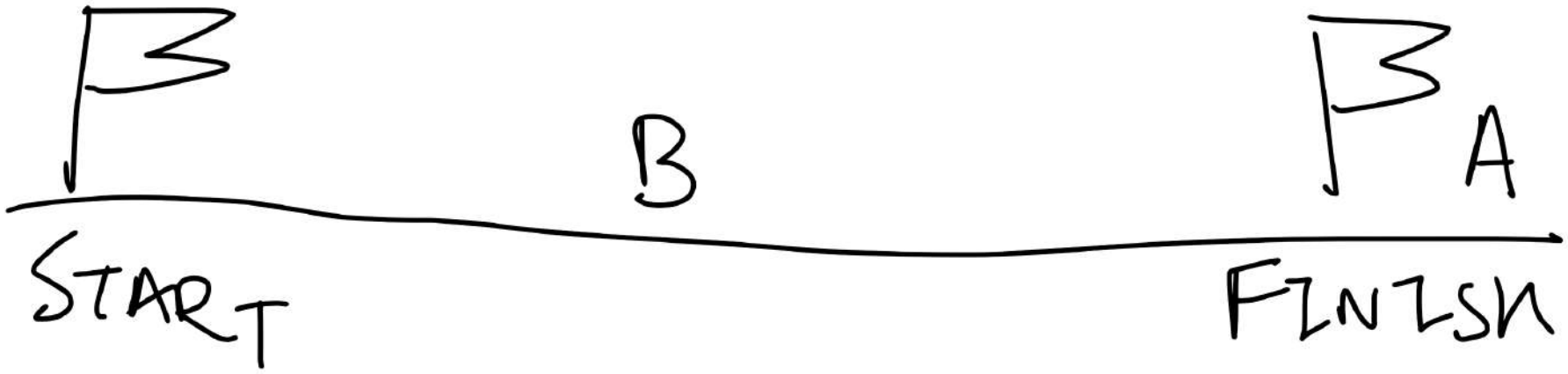
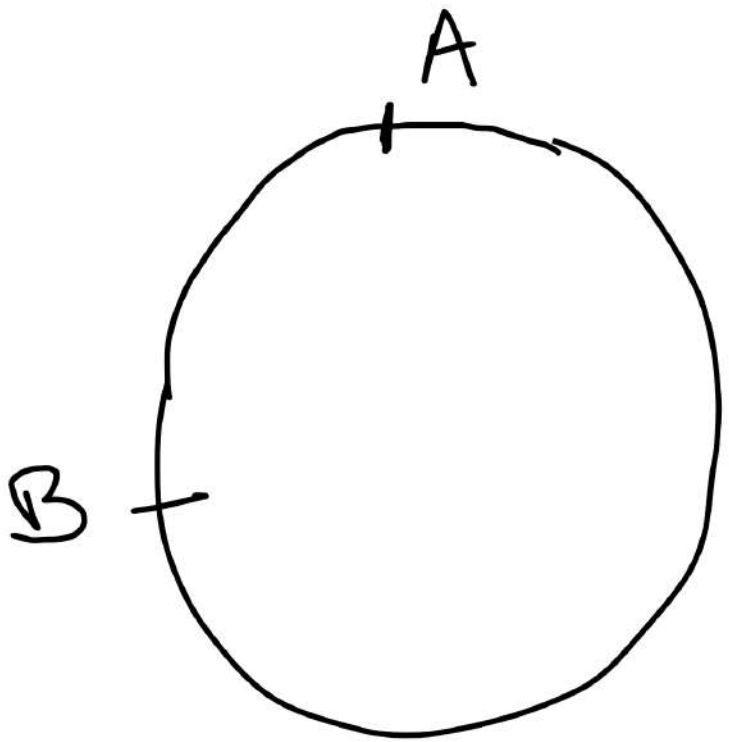
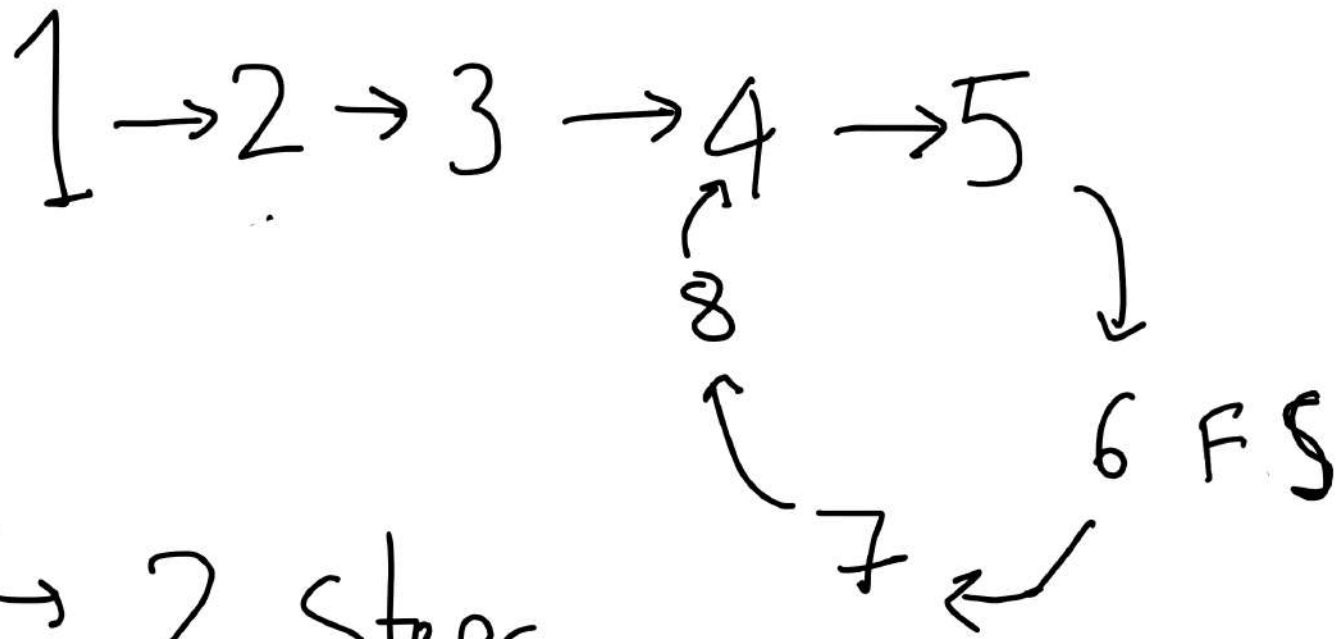

1 → 2 → 3 → 4
5
6
7
8
CYCLE
EXISTS IN
THIS LINKED LIST

HOW TO DETECT CYCLE?



Speed of A = 2 x Speed of B


A
B



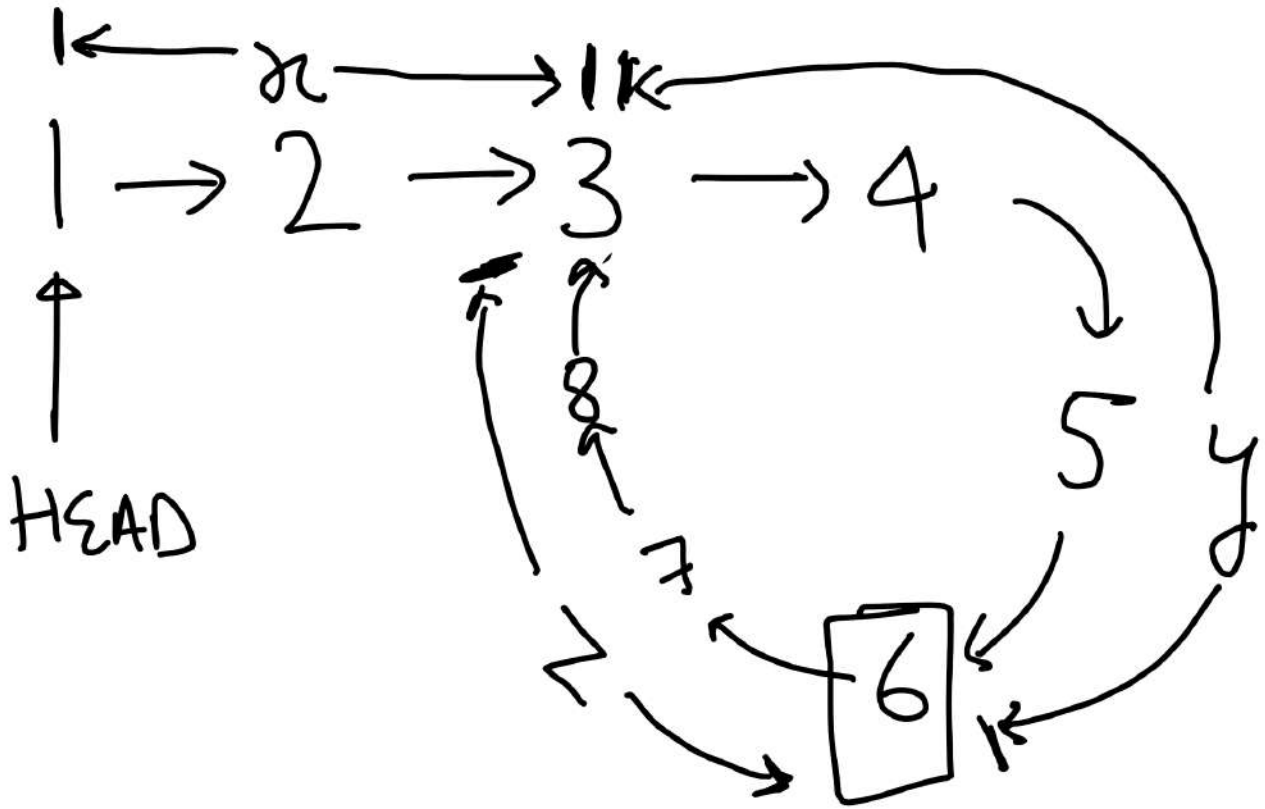
F → 2 steps

S → 1 steps

S	F
2	3
3	5
4	7
5	4
6	6

IF SLOW & FAST POINTERS END UP MEETING AT SOME POINT, THEN IT MEANS A CYCLE EXISTS.

FLOYD CYCLE DETECTION



Distance travelled by Slow = $x + y$

Distance travelled by Fast = $x + 2y + z$

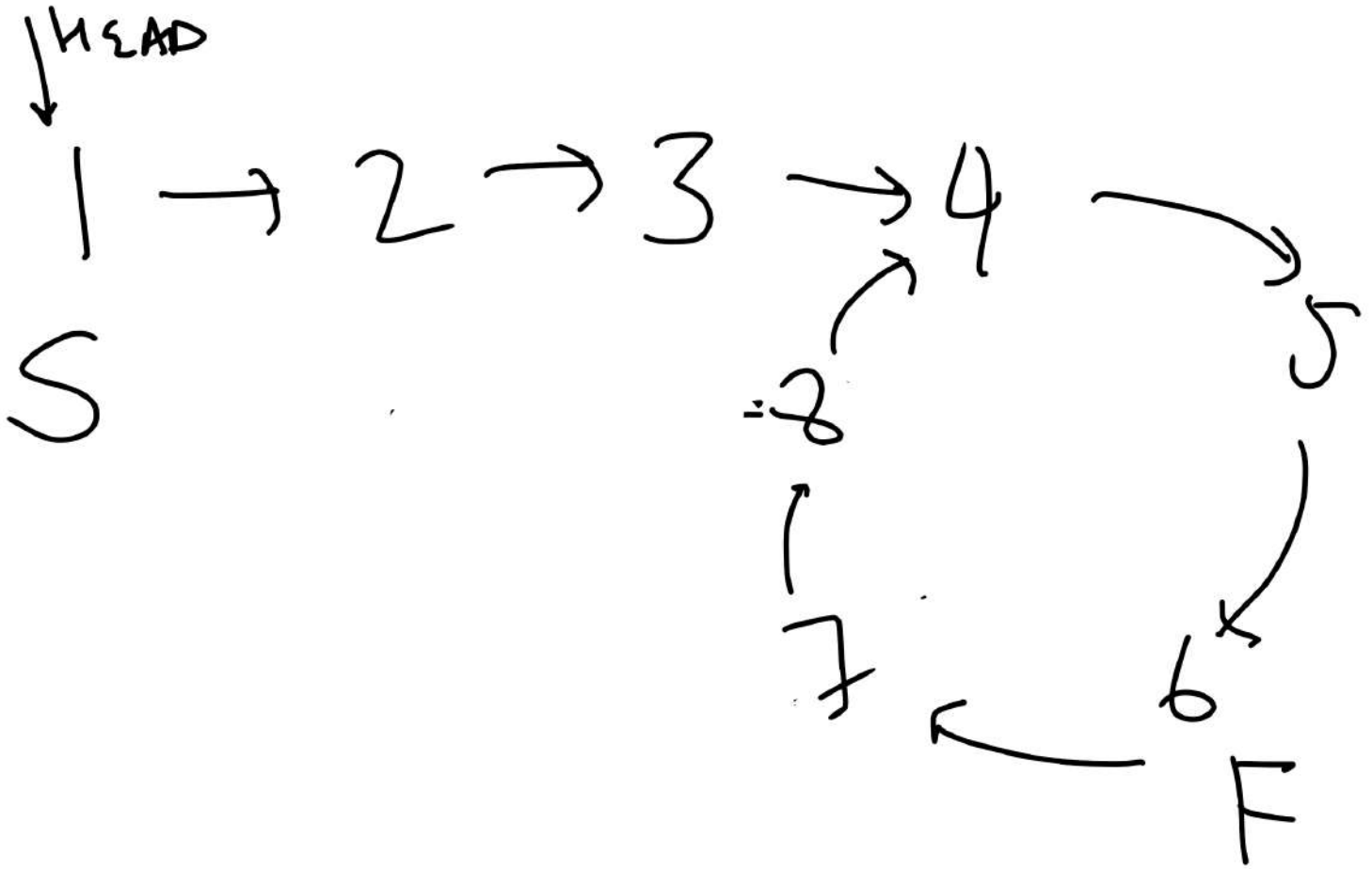
Distance travelled by Fast = $2 \times$ (Distance travelled by Slow)

$$x + 2y + z = 2(x + y)$$

$$x + 2y + z = 2x + 2y$$

$$x + z = 2x$$

$$\Rightarrow \boxed{x = z}$$


HEAD
1
2
3
4
5
6
7
8
S
F

MOVE ANY POINTER SLOW OR FAST TO HEAD. LET'S SAY YOU ARE MOVING SLOW TO HEAD, THEN KEEP FAST AT MEETING POINT ONLY.

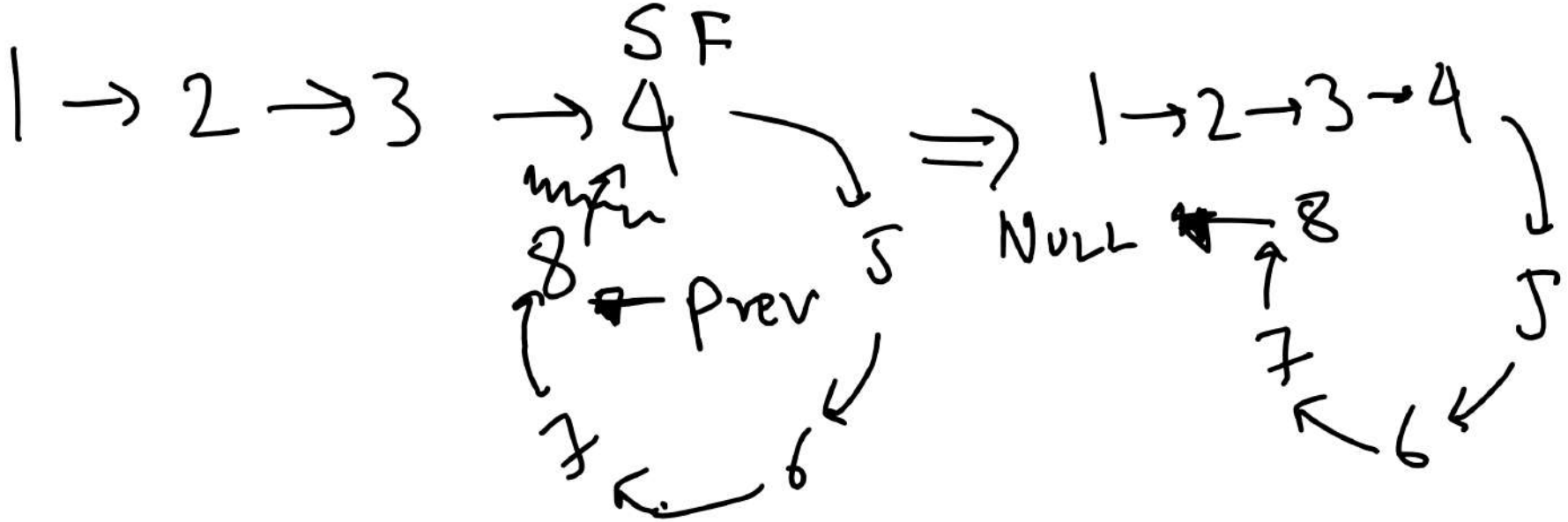
Now move slow & fast by one step only, i.e.

slow = slow.next

fast = fast.next

slow & fast will meet at the starting point of the cycle.

NOW, JUST SET THE NEXT OF PREV NODE TO NULL. FOR INSTANCE -



1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → NULL