Put two pointers on the list:

Slow moves one node at a time.

Fast moves two nodes at a time.

2.

Let them walk until they land on the same node. If they never meet, there is no loop.

3

Once they do meet, remember that node.

From this meeting point to the first node of the loop, the number of steps is the same as from the head of the list to that first loop node.

4

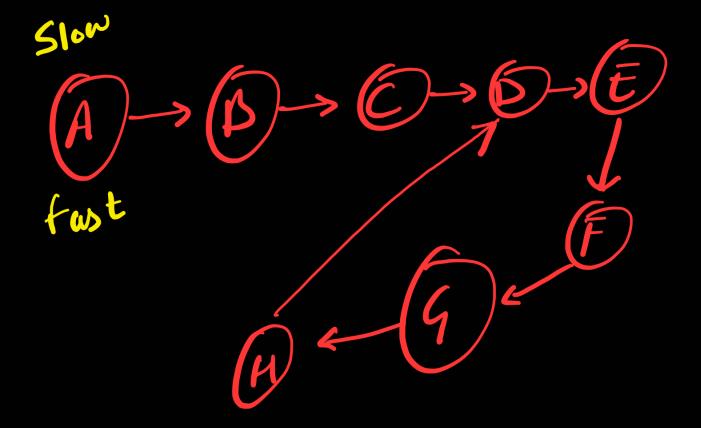
Now take two new pointers:

One starts from the head of the list.

The other starts from the meeting point of the slow and fast pointers.

Move both pointers one step at a time.

The node where they meet is the starting point of the cycle.



A when we do cycle detection.

Slow	fust	
A	A	
B	C	
<u></u>	E	
D	4	
E	D	
F	F	4 meets

Distance (A-,D) => 3 Distance (F > D) => 3

