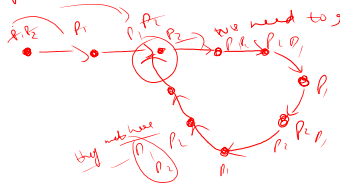
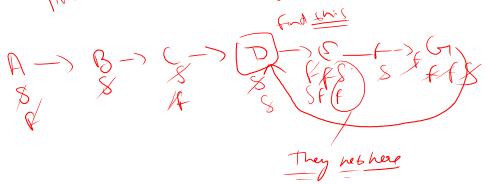
Loop detection in linked list



→ This can be done if we modify the list & update some unique no. and then when the no. is encountered that is the beginning of loop.

→ We can also use hashmap and then check if it has been encountered before or not.

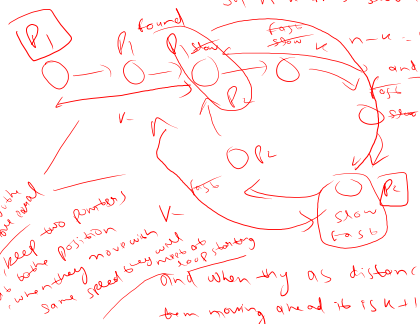
→ Something came into mind if we keep a pointer that moves with speed 1 & another with 2 if they meet at any point then there is a cycle



Distance travelled by S = n
f = 2n

When slow entered they were k (assumed) nodes apart and then k+1, ... k+2, and when it reached n (no. of nodes in loop) they were at same position.

So, n-k times slow had to move forward.



And when they as distance b/w the m is K, then moving ahead it is K+1, K+2, ... K+n → they meet

When they are n distance apart, So, n = n - K,

they as slow had to move n-k steps before colliding