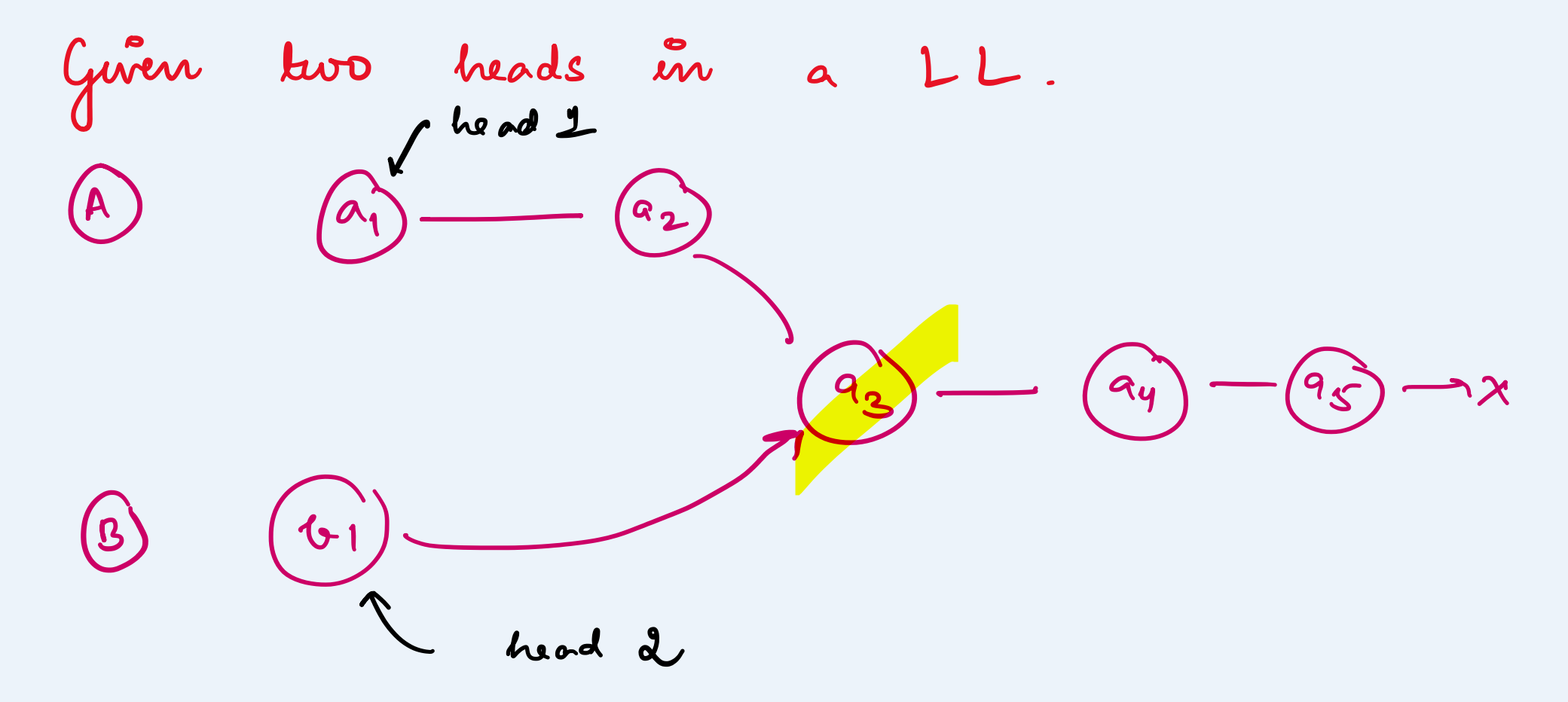
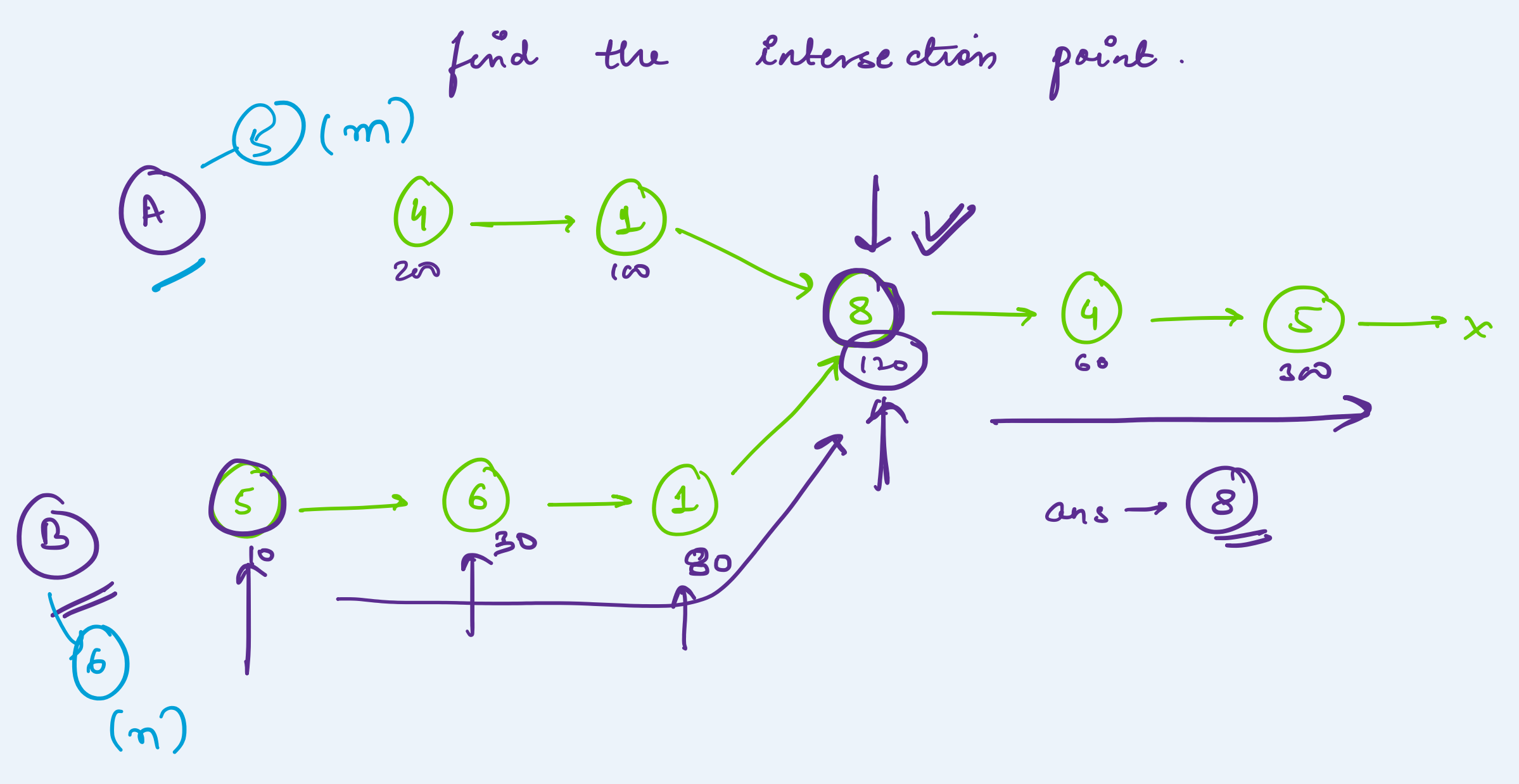
Thursday, 23 September 2021 8:28 PM

Ques Given two heads in a LL.



find the intersection point.



Approach ①

→ Store all the address of nodes of linked list L1 in a set.

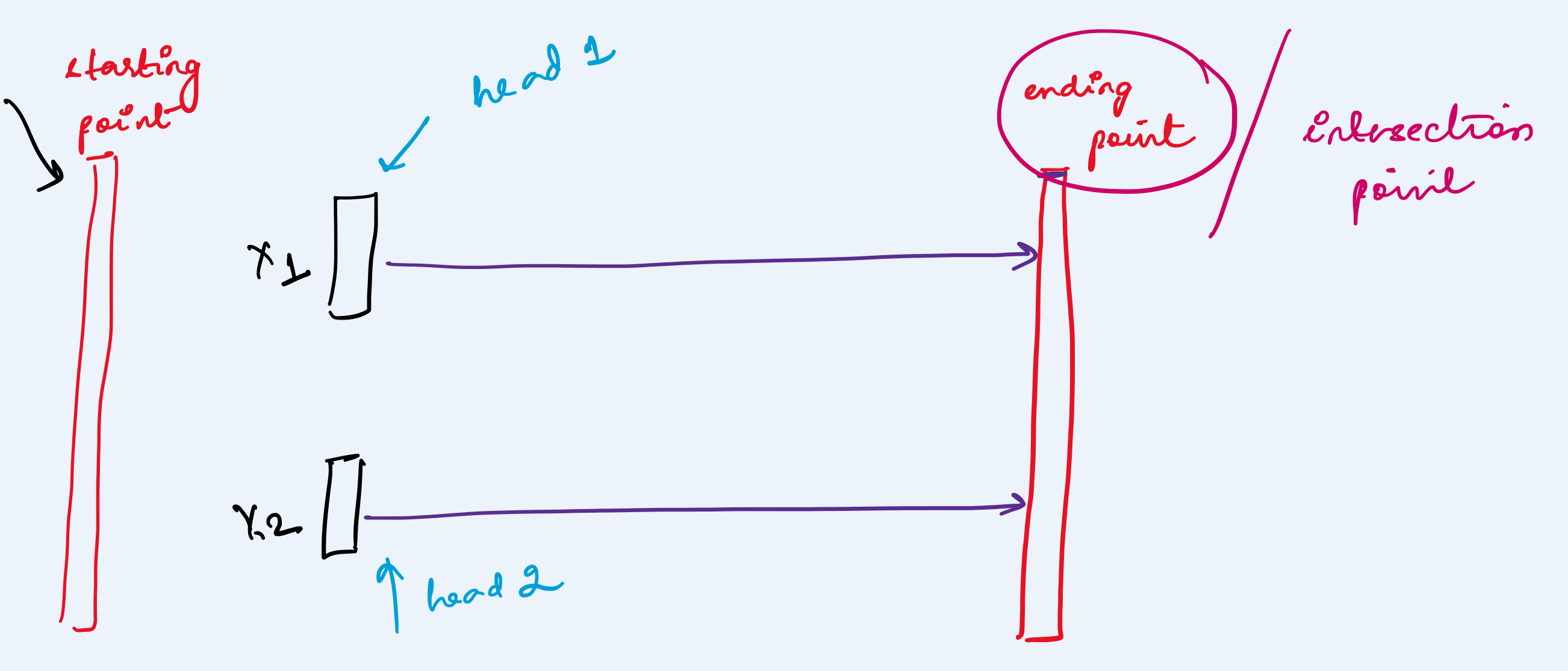
→ < 200, 100, (120), 60, 300 >

→ Iterate the second linked list L2 and check if any node address is present in the set.

TC → $O(n+m)$

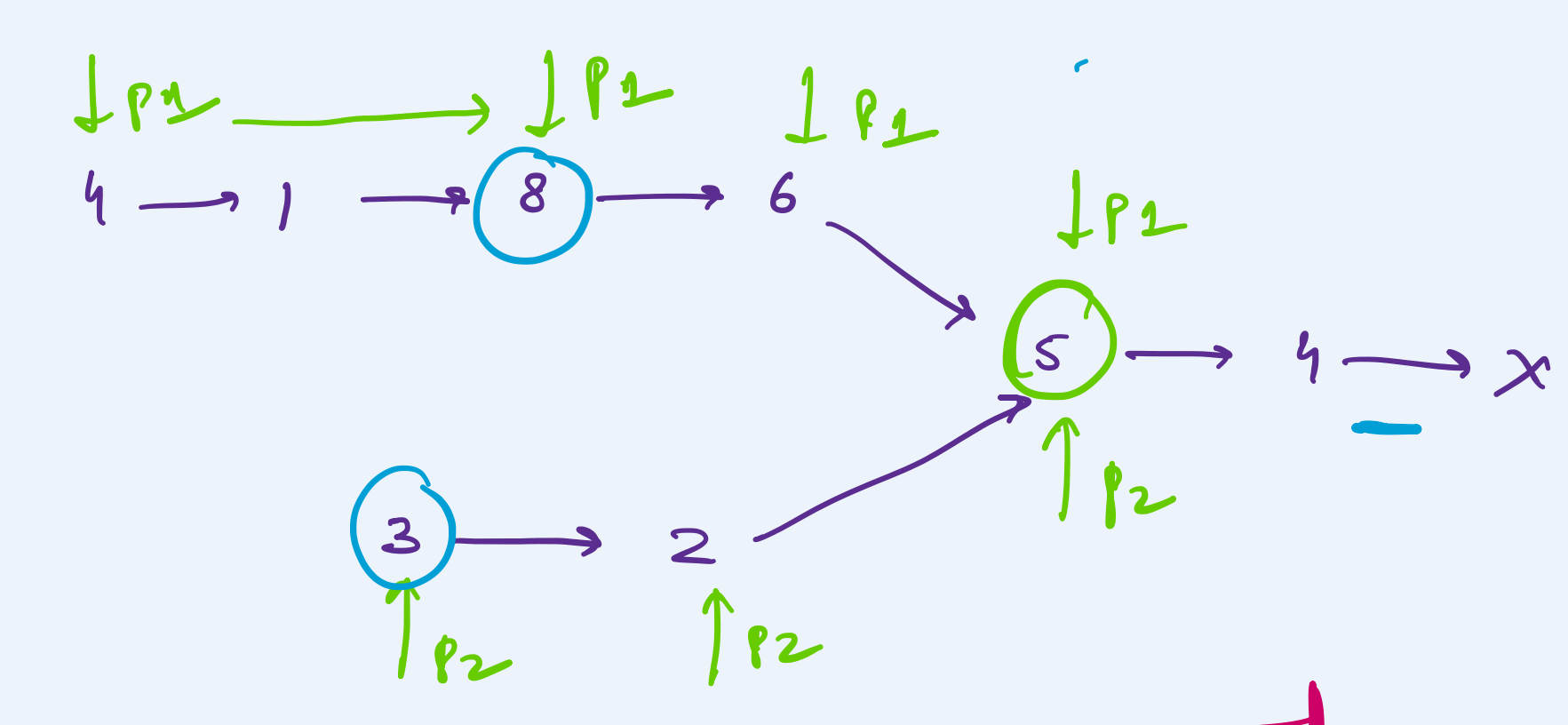
SC → $O(n) / O(m)$

Approach ② :



speed (x1) = speed (x2)

Race starts at t = 0.



① Find len (l1) = 6

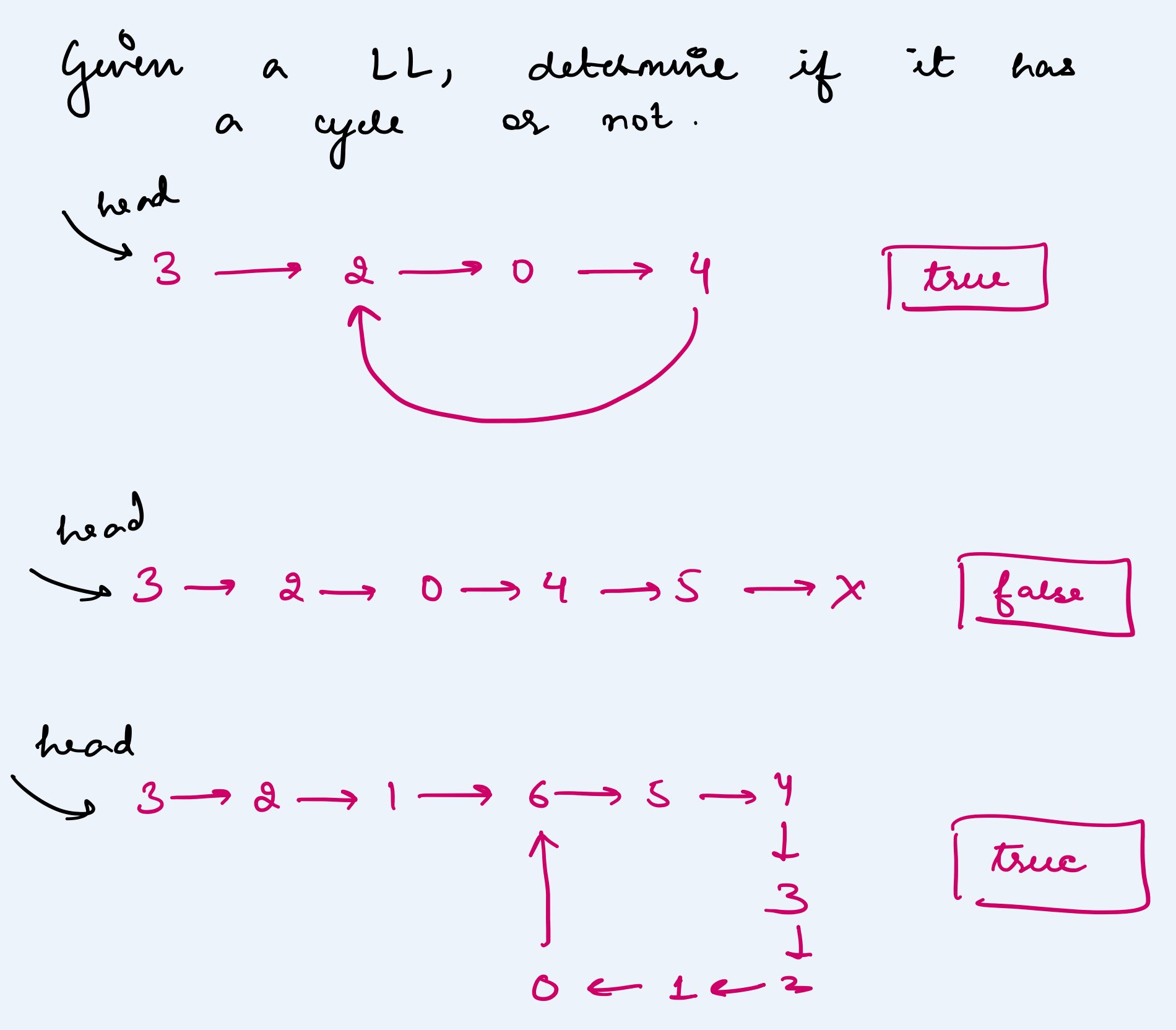
Find len (l2) = 4

```
while (p1 != p2) {
    p1 = p1.next;
    p2 = p2.next;
}
```

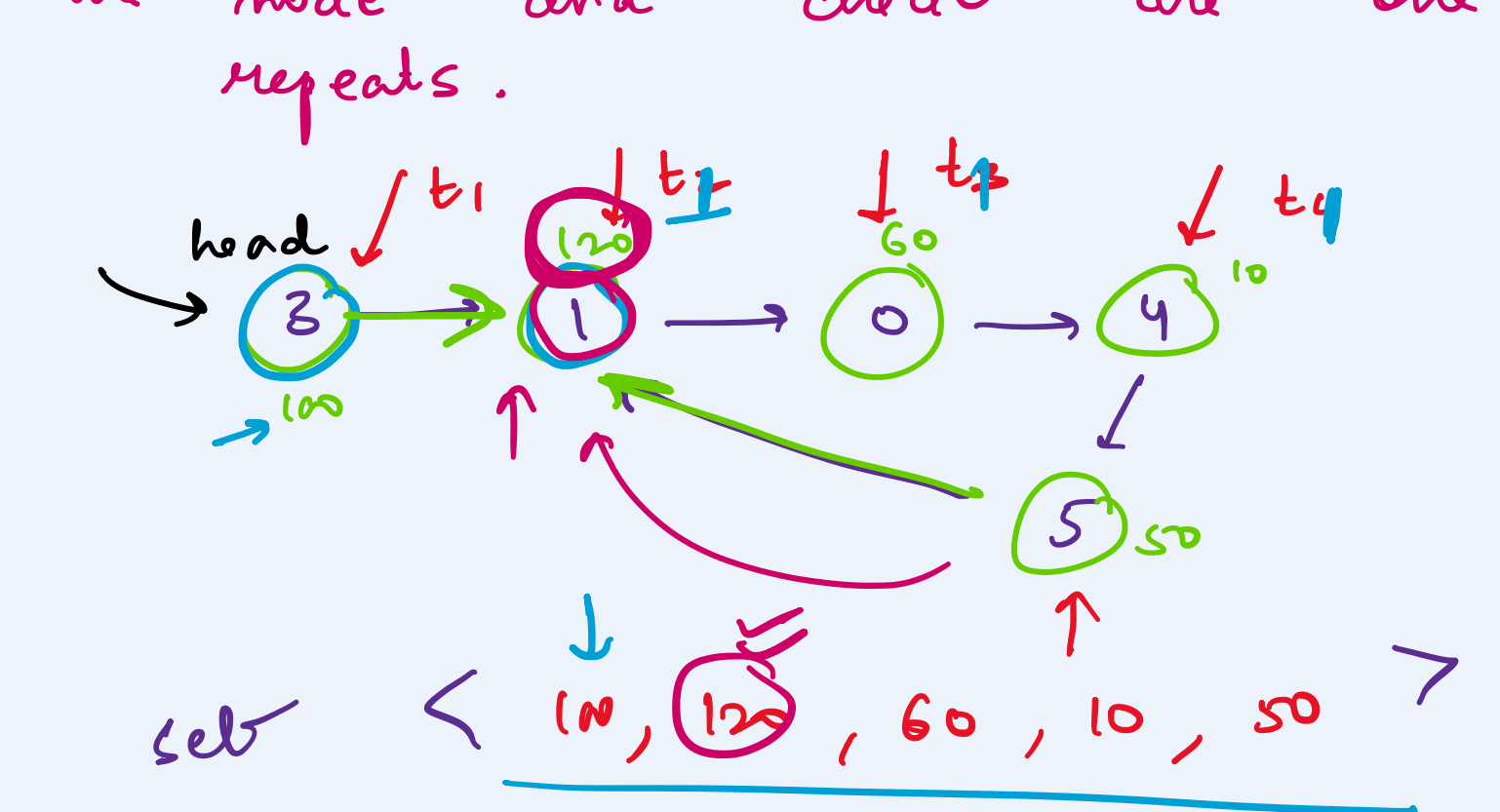
SC → $O(1)$

TC → $O(n+m)$

Q Given a LL, determine if it has a cycle or not.



Approach ① : Store the addresses of the node and check the one which repeats.

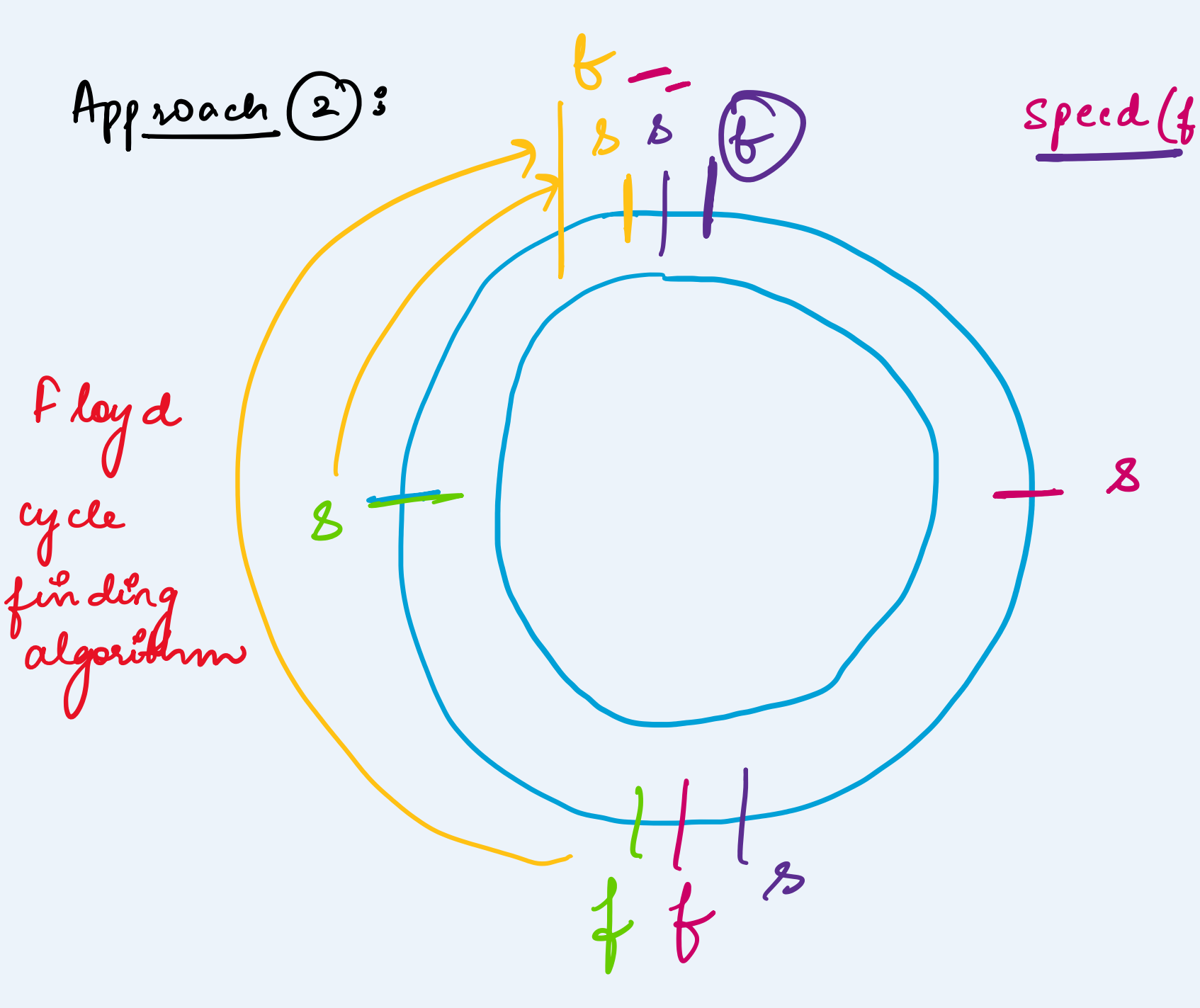


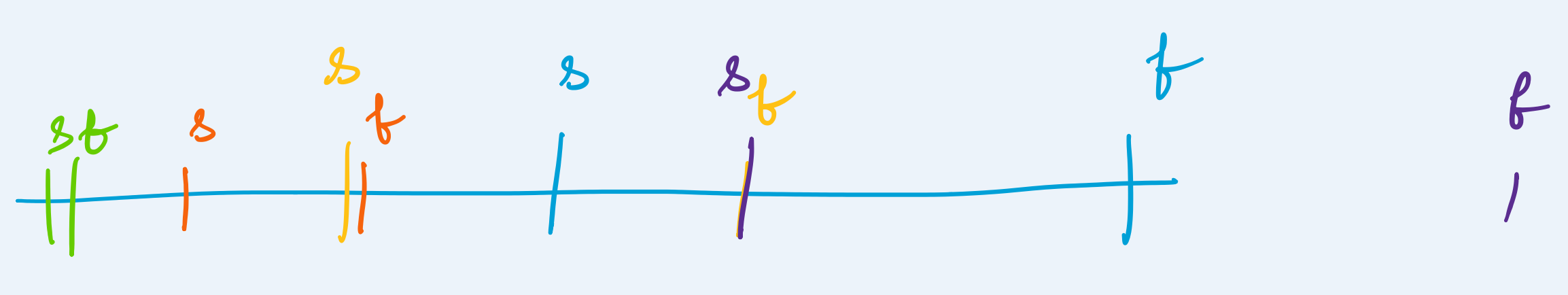
TC → $O(n)$

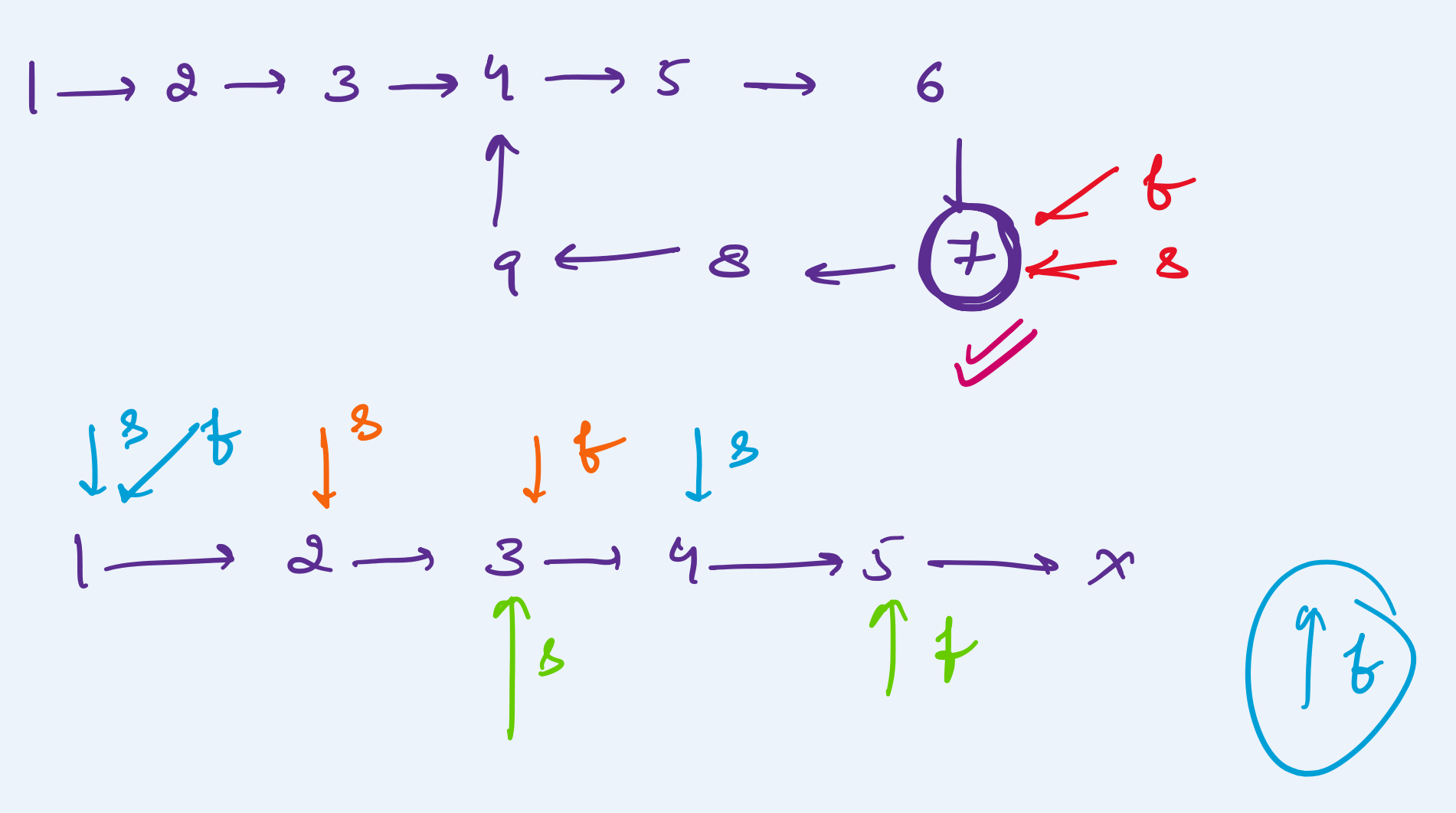
SC → $O(n)$

Approach ② :

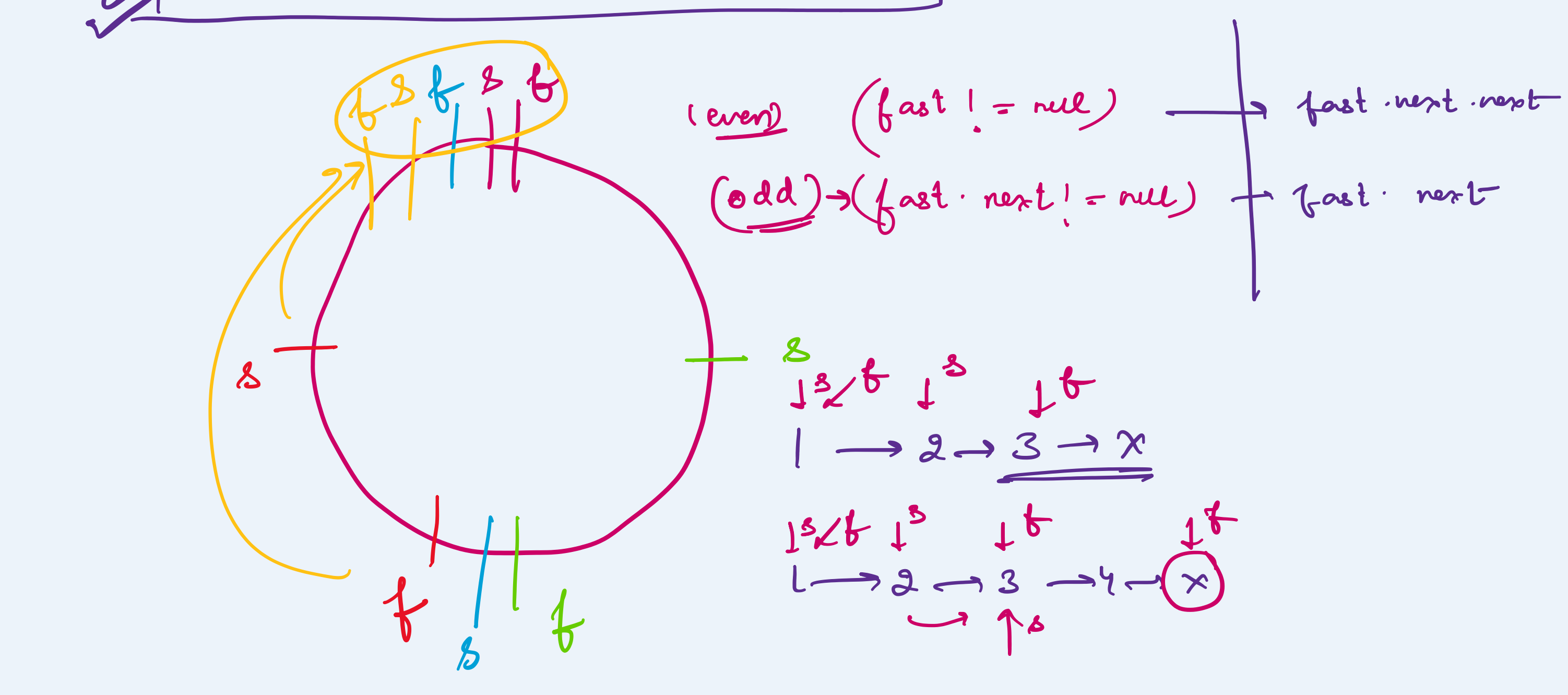
speed(f) = 2 * speed(s)





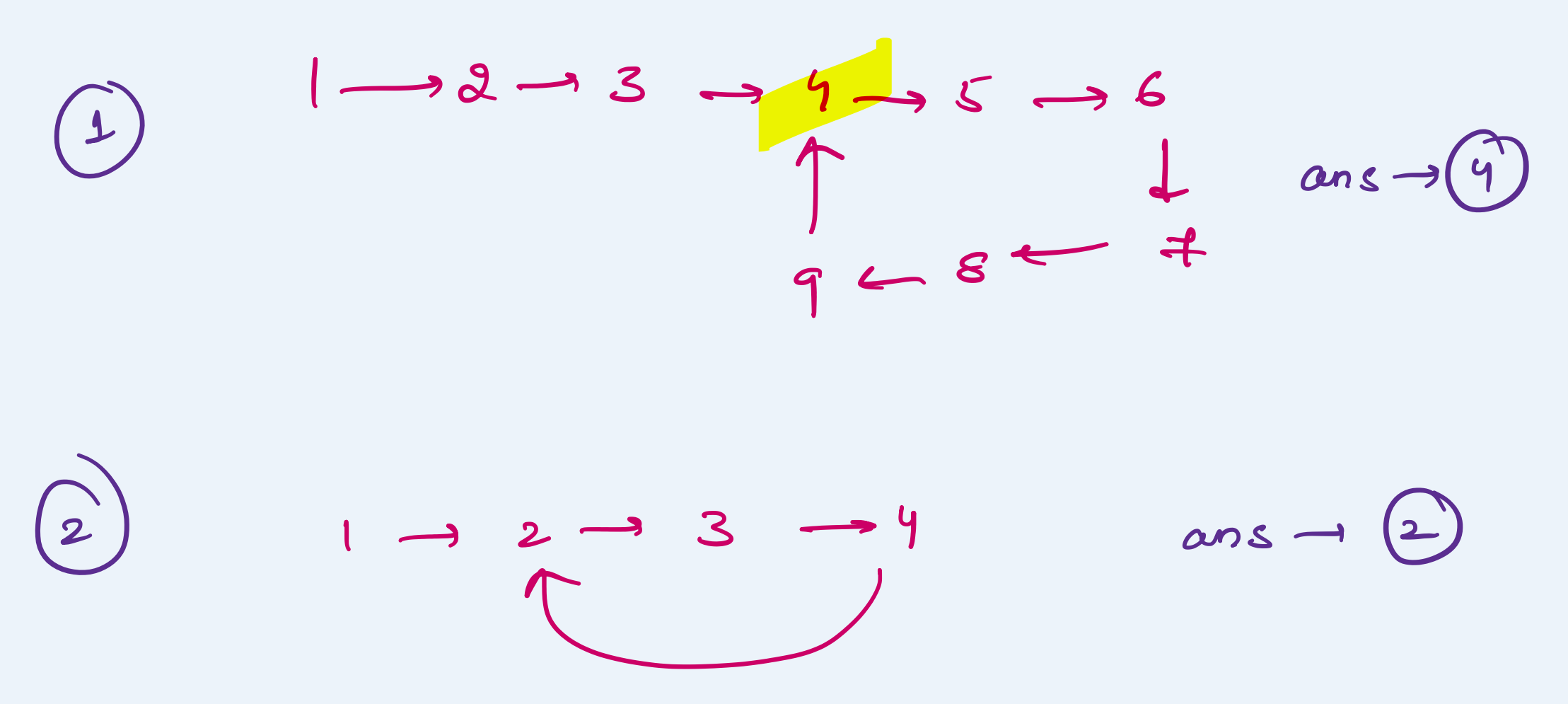


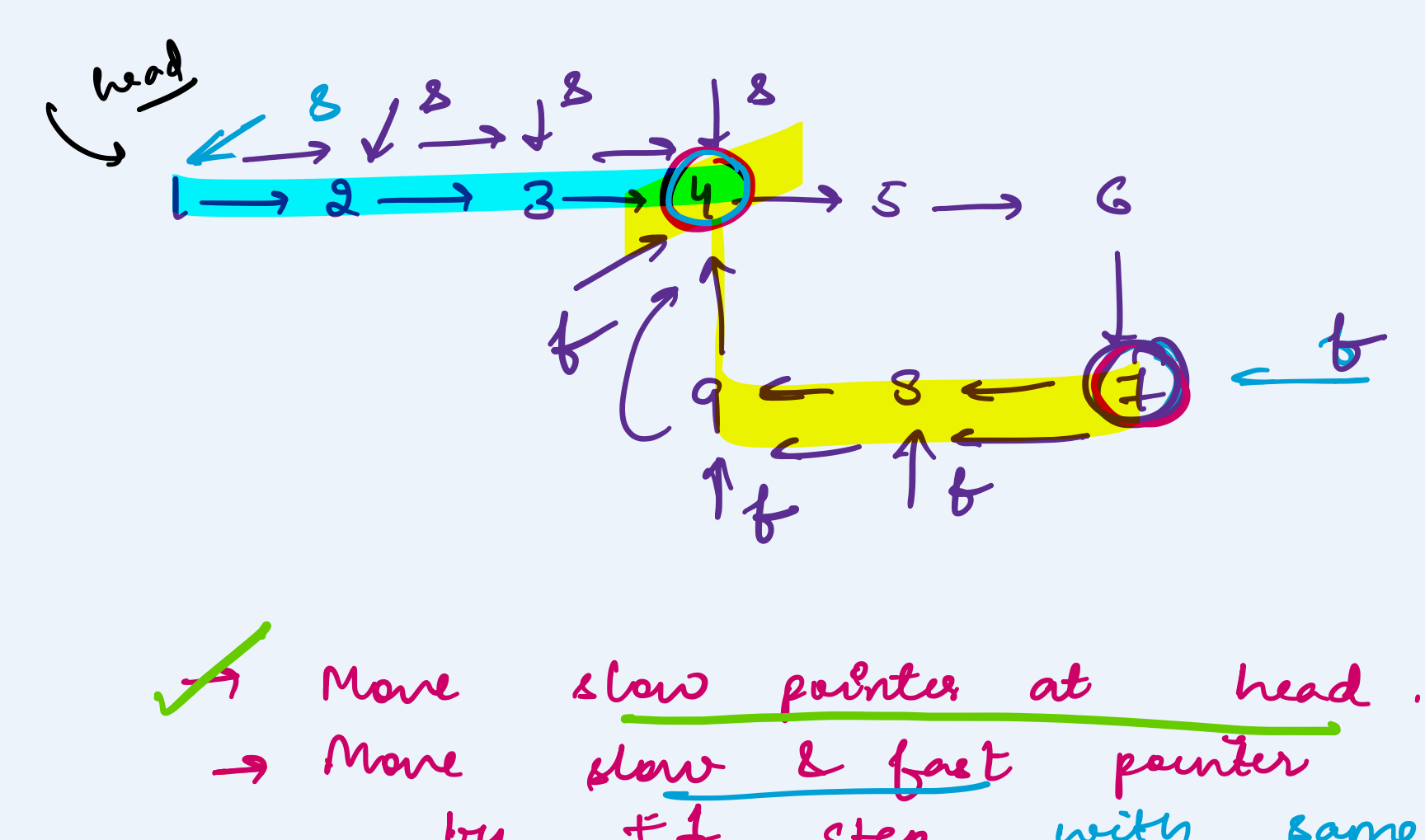
3 * speed (s) = speed (f) ??



Q If there is a cycle? (Yes)

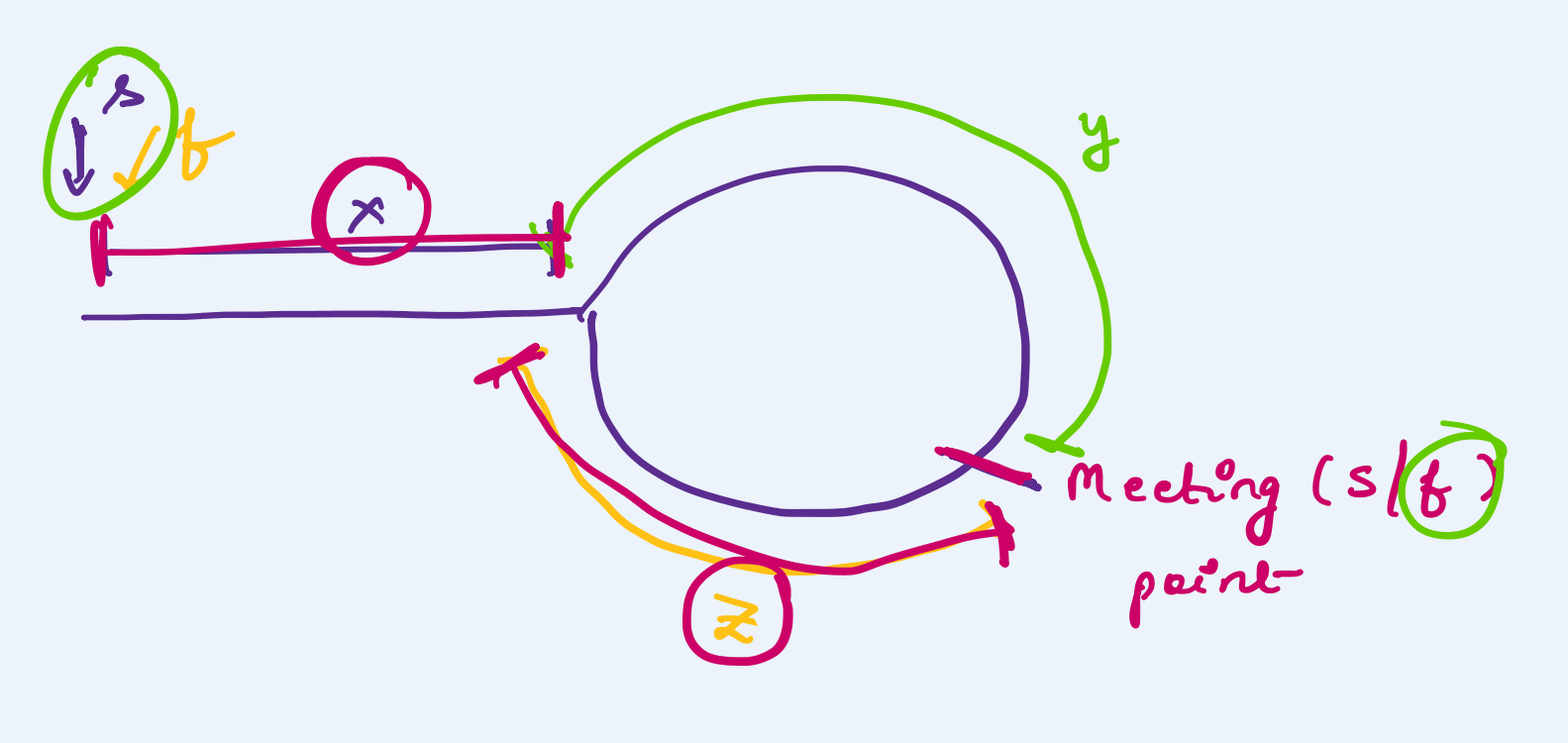
Then, identify the starting point of the cycle.





→ Move slow pointer at head.

→ Move slow & fast pointer both by +1 step, with same speed



Distance travelled by [slow] before coming at meeting point = $x+y$

Distance [fast] $= x + y + z + y$

$= x + 2y + z$

speed (fast) = 2 * speed (slow)

$S = \frac{D}{T}$

$S_s = \frac{D_s}{T_s}$ and $S_f = \frac{D_f}{T_f}$

$T_s = T_f$

$\frac{D_s}{S_s} = \frac{D_f}{S_f}$

$\frac{D_s}{S_s} = \frac{D_f}{2 * S_s}$

$D_f = 2 D_s$

2 * Dist (slow) = Dist (fast)

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

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

$2x - x = z$

$x = z$