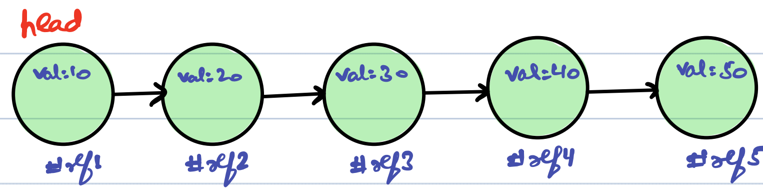
Today's agenda

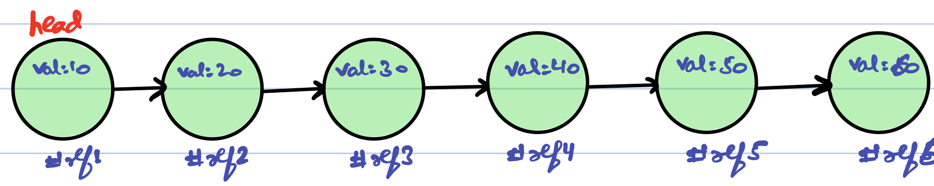
↳ Find mid

↳ floyd cycle

Q) Given head of a linkedlist, find the mid of it.



→ ans=30



→ ans=30

//idea1

↳ iterate and find out length of linkedlist and then iterate again to return $\frac{length}{2}$ element.

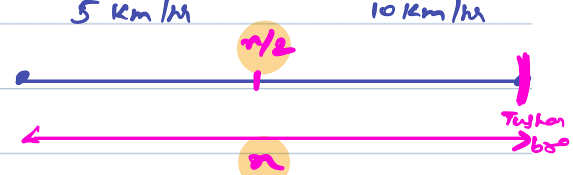
T.C: $O(n + n/2)$ S.C: $O(1)$

$= O(n)$

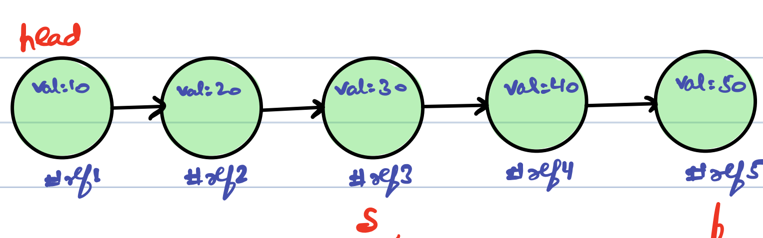
//idea2 → Single iteration

Sankeerth ↓ 5 km/hr

Tushar ↓ 10 km/hr

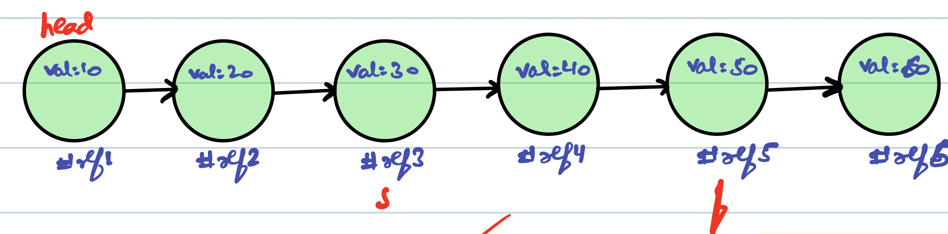


odd length

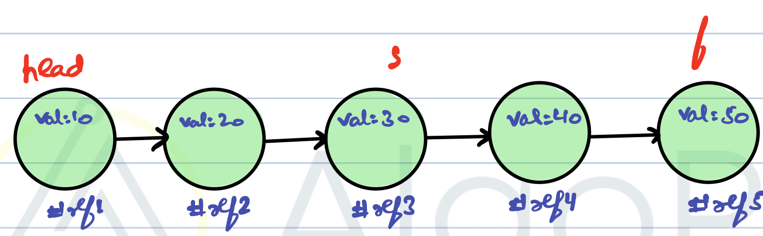


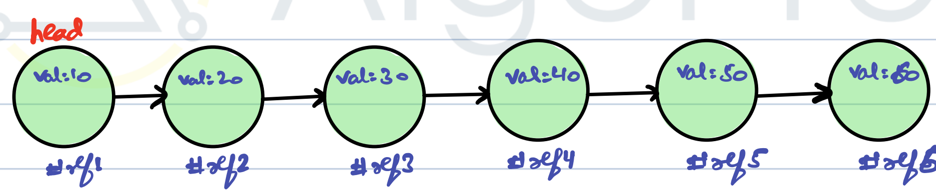
f.next == null → stop → s is our answer

even length



if f.next.next == null → exit → s is our ans.





//Psuedo code

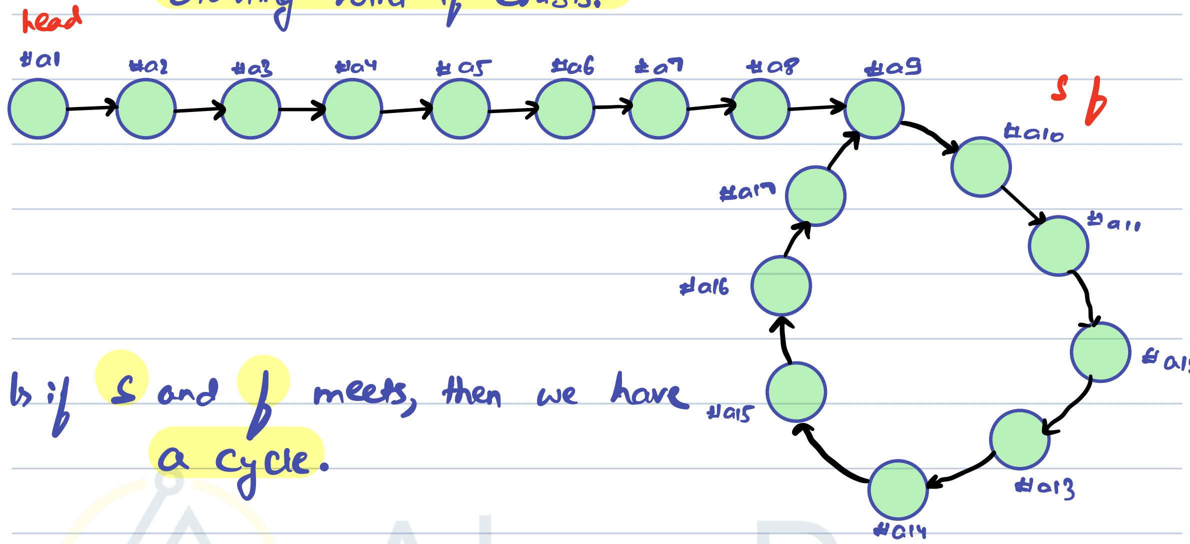
```
Node mid (Node head){

        Node s = head;
        Node f = head;

            while (f.next != null && f.next.next != null) {
                s = s.next;
                f = f.next.next;
            }

            return s;

}
```

T.C: O(N)

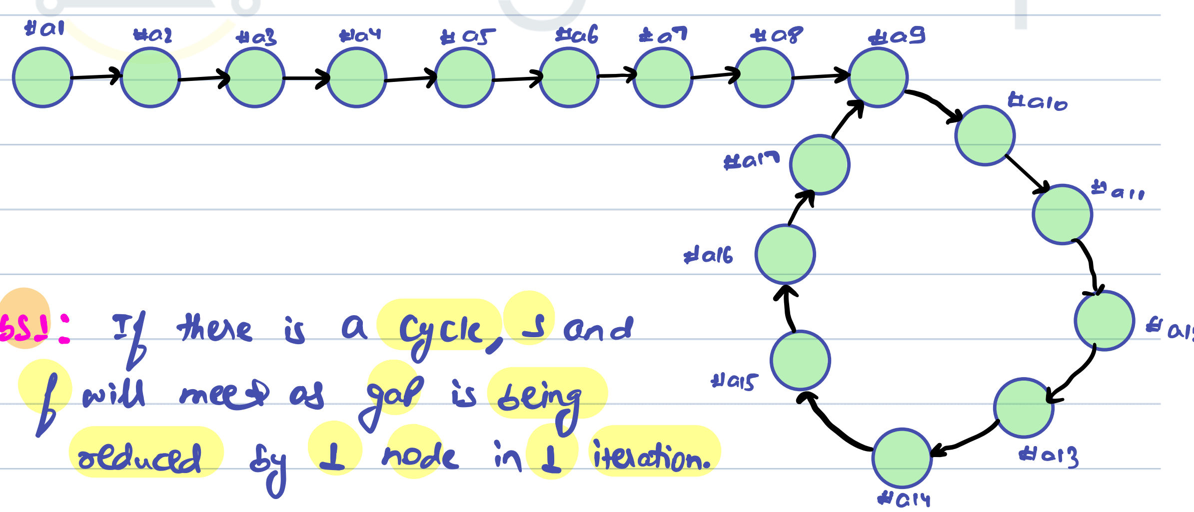
S.C: O(1)

floyd cycle

Q) Given a linked list, check for cycle & return the Starting Point if exists.



↳ if s and f meets, then we have a cycle.

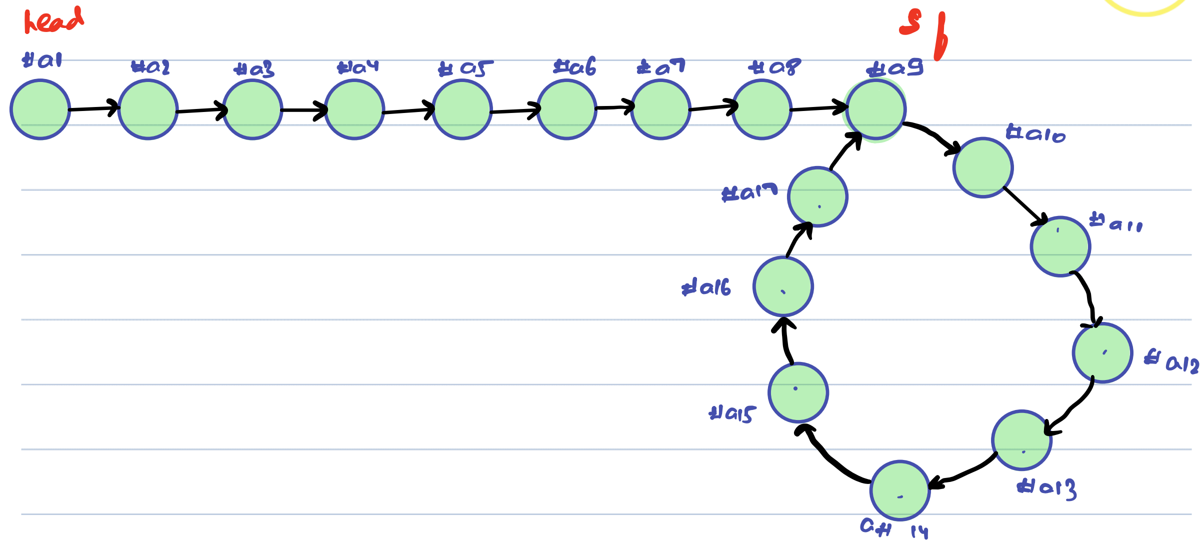


OBS1: If there is a cycle, s and f will meet as gap is being reduced by 1 node in 1 iteration.

OBS2:

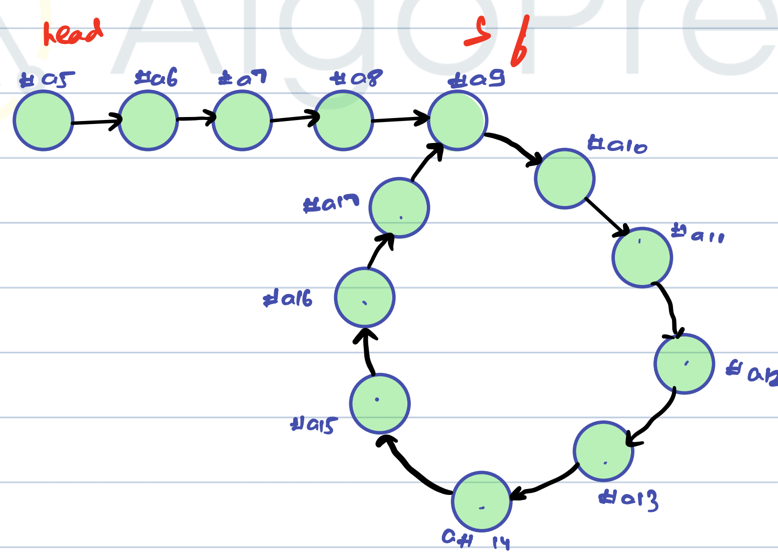
↳ slow can never complete the cycle.

Ex1:



Ex2:

Chain : 4   Cycle : 9



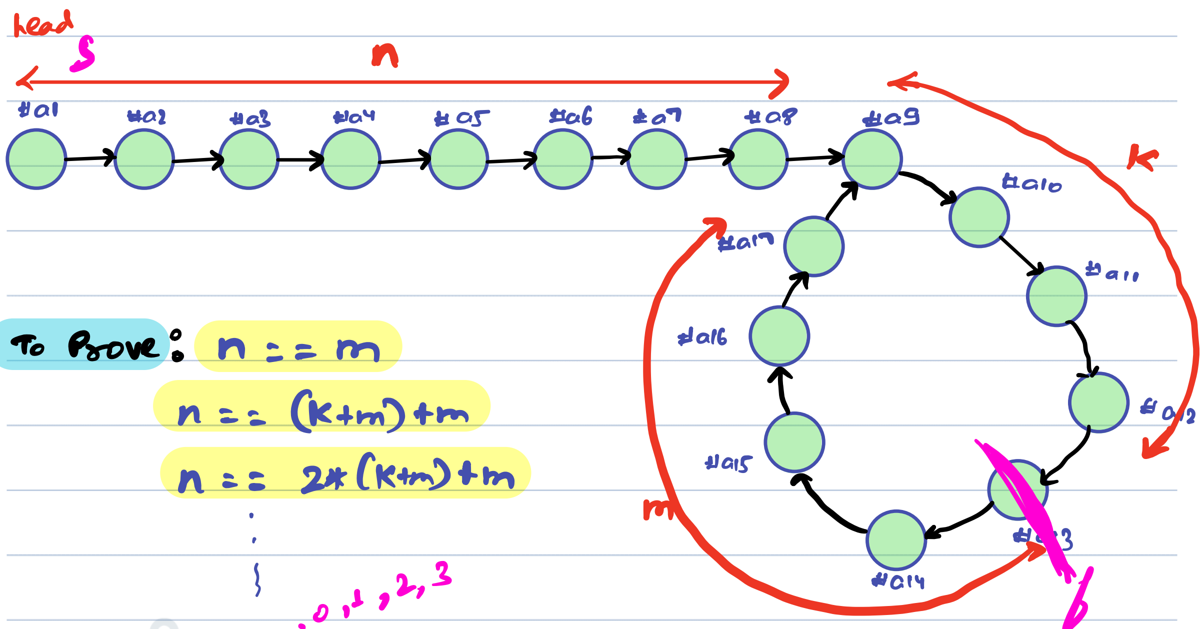
//Psuedo code

```
Node removeCycle (Node head) {
    Node s = head;
    Node f = head;

    while (s != f) {
        f = f.next.next;
        s = s.next;
    }
    s = head;

    while (s != f) {
        s = s.next;
        f = f.next;
    }

    return s;
}
```

T.C: O(N)

S.C: O(1)

Why??



To Prove: $n == m$

$n == (k+m)+m$

$n == 2*(k+m)+m$

$\vdots$

0, 1, 2, 3

$n = x*(k+m)+m$

$n = (c-1)*(k+m)+m$

Proof:

$slow = n+k$

$fast = n + c*(k+m) + k$

$2*slow = fast$

$2*(n+k) = n + c*(k+m) + k$

$2*(n+k) - (n+k) = c*(k+m)$

$n+k = c*(k+m)$

$n+\cancel{k} = (c-1)*(k+m) + (\cancel{k}+m)$

$n = (c-1)*(k+m) + m$

↳ Hence Proved