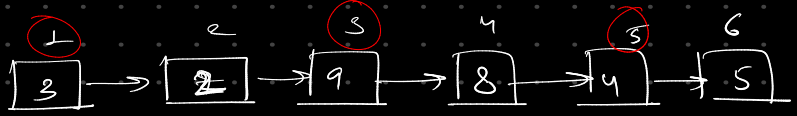
Odd & Even Index

Segregate odd & even indices in a LL

Eg:



Brute fore approach :

↳ Traverse through odd places using temp.next.next & store the values in a list.

(N/2)

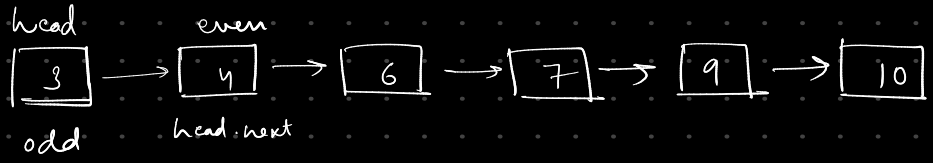
↳ Traverse through even places using temp.next.next and store the values in a list.

N/2)

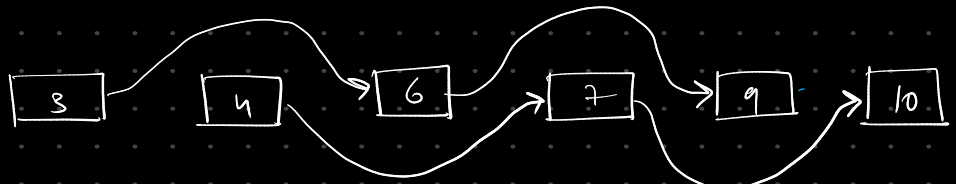
⇒ Overwrite the linked list using the new list created & return head.

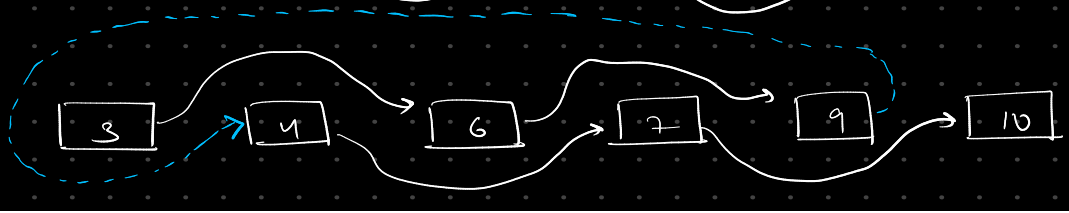
TC : O(2N) & SC: O(N)

Optimal Approach (change the links) :



⇒





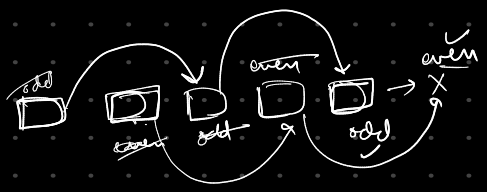

```
Node odd = head, even = head.next,

while (even != null && even.next != null) {

    odd.next = odd.next.next;

    odd = odd.next;

    even.next = even.next.next;

    even = even.next;

}

// Now joining both sub lists.

odd.next = evenHead;
```

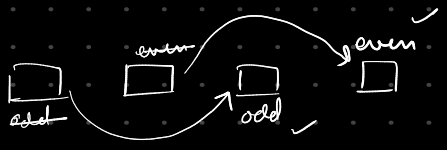
evenHead = head.next.

↳ store the value of even head

⇒ will be used later.

∵ even is ahead of odd in the starting, ∴ even will be the deciding factor for while loop cond^n.





↳ because head.next is not what it originally was.

↳ so even head will be the head of even list as the original value of head.next was memorized in it.

$TC \leq O(N)$

$SC = O(1)$