

Palindrome Linked List

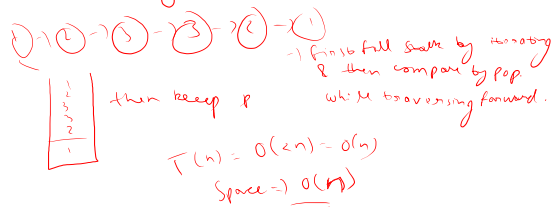


→ Here what we can do is
 check first element and last element, if they
 match remove both and then move ahead
 and check again till lists will be left with single
 element.

→ Here $T(n) = O(n^2)$
 because $\rightarrow n + n-2 + n-4 + \dots + n-2$
 $\rightarrow (n+1)n + (2+n) + \dots + (n)$
 \rightarrow and $n = \frac{n}{2}$ so, $O(n^2)$
 Space $\rightarrow O(1)$

→ This will not work as we will have to keep
 a previous pointer to remove the tail.

It can be done using stack.



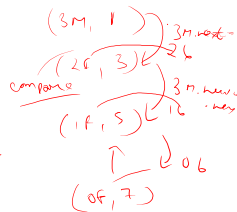
→ can we do it with recursion.

→ We can also do it with reversing the linked list then
 matching. It will also be $O(n)$ & space $O(n)$.

0 → 1 → 2 → 3 → 2 → 1 → 0 → we need to know length to
 reach middle.

recursion (Node n, int length):

Base case (length = 0 || length == 1);



→ we also need to return
 true or false if matched or
 not so we need wrapper class.