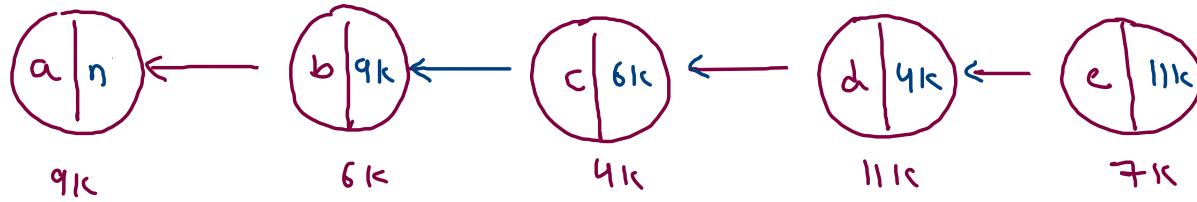


Reverse LL

type	space	time
①. pointer iterative	$O(1)$	$O(n)$
②. data iterative	$O(1)$	$O(n^2)$
③. pointer recursive	$O(n)$	$O(n)$
④. data recursive	$O(n)$	$O(n)$



```
Node {  
    int val;  
    Node next;  
}
```

head = 9k

```
n = c.next  
c.next = p  
p = c  
c = n
```

```

public static ListNode reverse(ListNode head) {
    ListNode p = null;
    ListNode c = head;

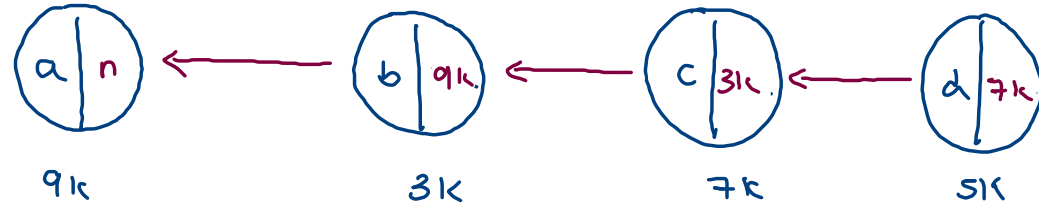
    while(c != null) {
        //preserve
        ListNode n = c.next;

        //connection
        c.next = p;

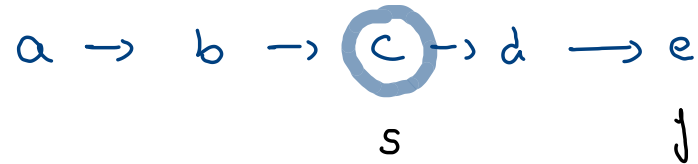
        //move
        p = c;
        c = n;
    }

    return p;
}

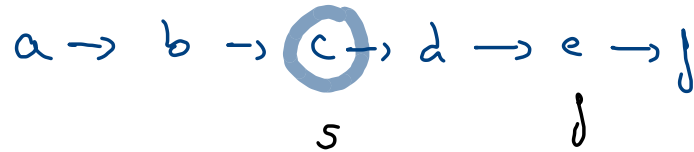
```



Middle Of A Linked List (portal) : If there are two mid nodes, return the first one.

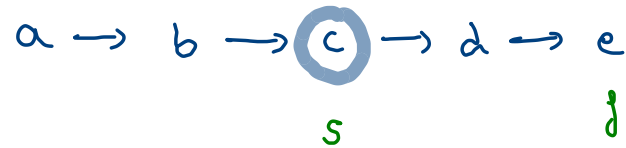


(n → odd) fast.next == null

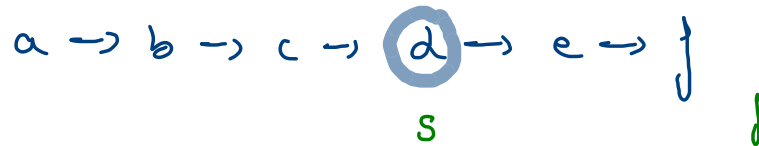


(n → even) fast.next.next == null

Middle Of A Linked List (LC) : If there are two mid nodes, return the second one.

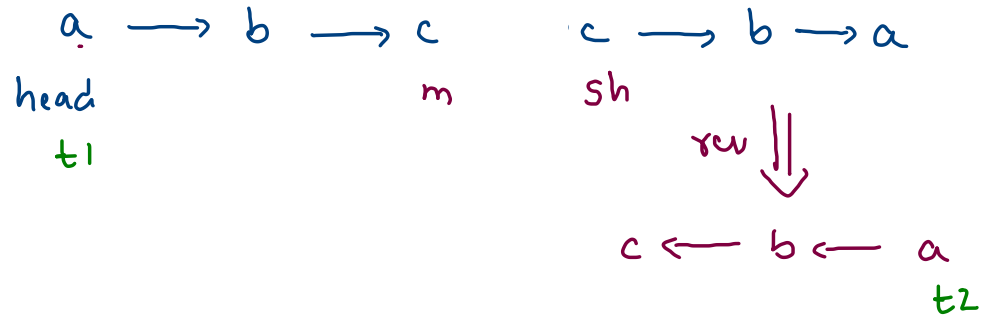


fast.next == null



fast == null

234. Palindrome Linked List



```

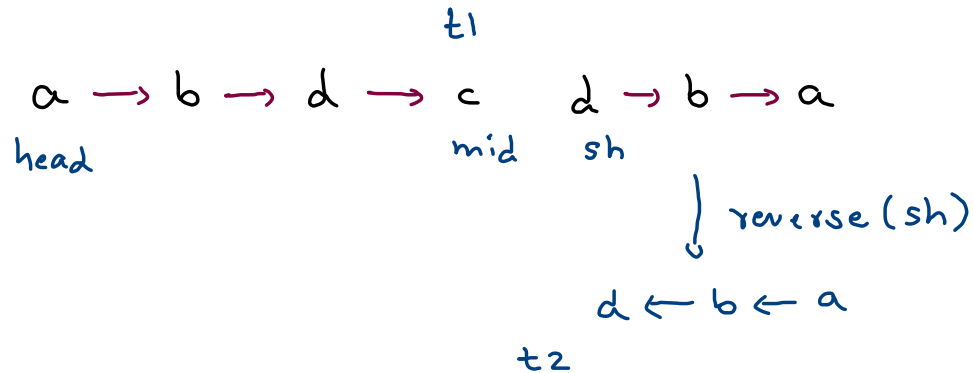
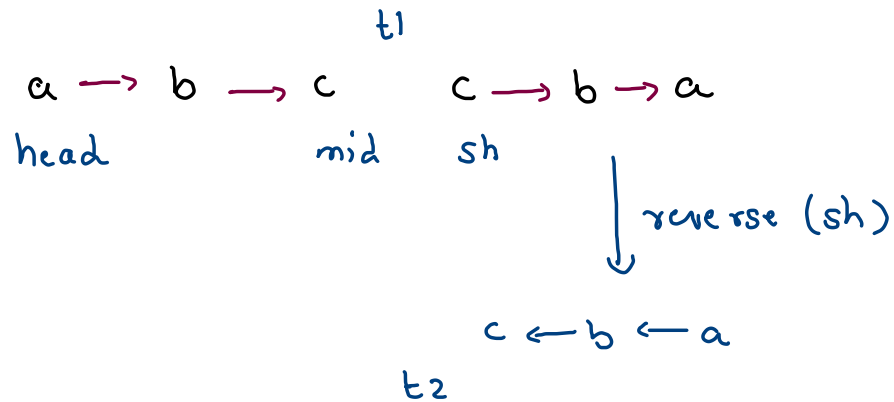
public boolean isPalindrome(ListNode head) {
    ListNode mid = mid(head);
    ListNode sh = mid.next; //second half head
    mid.next = null;

    ListNode t1 = head;
    ListNode t2 = reverse(sh);

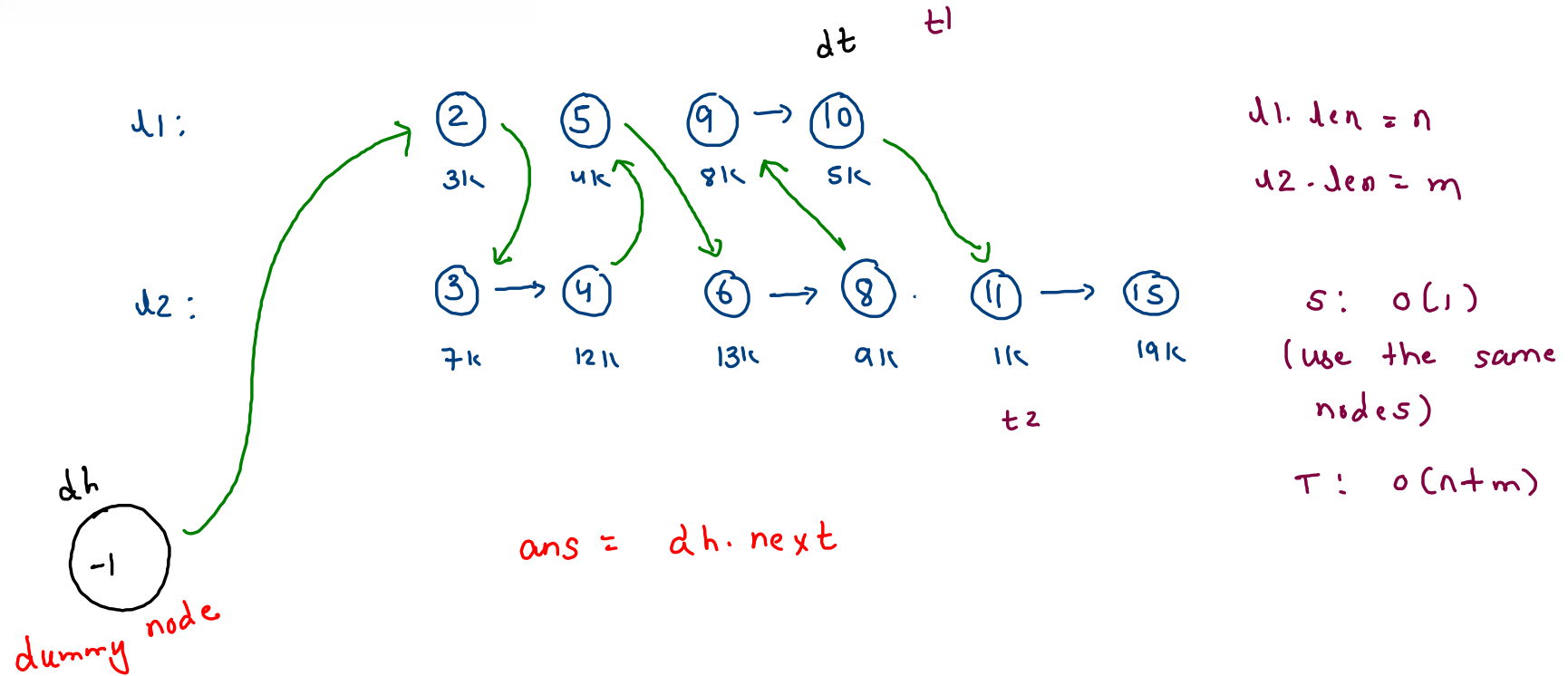
    while(t2 != null) {
        if(t1.val != t2.val) {
            return false;
        }
        t1 = t1.next;
        t2 = t2.next;
    }

    return true;
}

```



21. Merge Two Sorted Lists




```

public ListNode mergeTwoLists(ListNode list1, ListNode list2) {
    ListNode dh = new ListNode(-1);
    ListNode dt = dh;

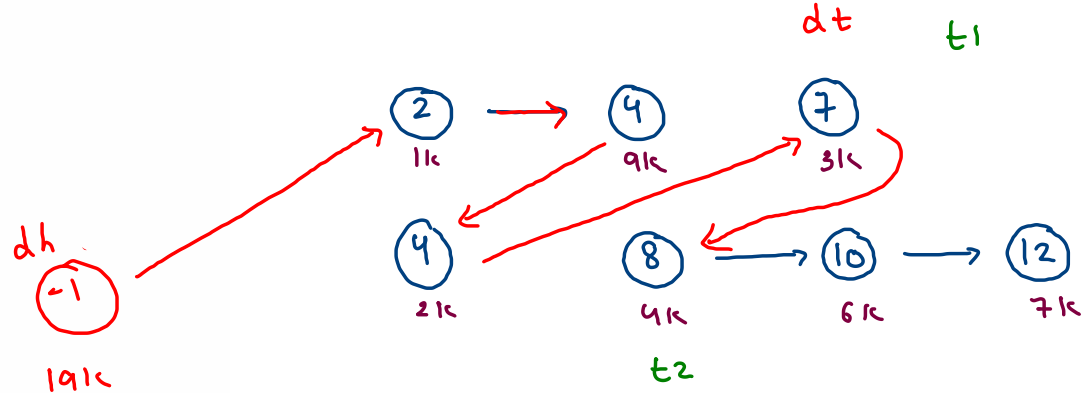
    ListNode t1 = list1;
    ListNode t2 = list2;

    while(t1 != null && t2 != null) {
        if(t1.val <= t2.val) {
            dt.next = t1;
            t1 = t1.next;
        }
        else {
            dt.next = t2;
            t2 = t2.next;
        }
        dt = dt.next;
    }

    if(t1 != null) {
        dt.next = t1;
    }
    else if(t2 != null) {
        dt.next = t2;
    }

    return dh.next;
}

```



ans : `dh.next = 1k`