**[Reorder List](https://leetcode.com/problems/reorder-list/)**

**class Solution {**

**public void reorderList(ListNode head) {**

**if(head == null || head.next == null) {**

**return;**

**}**

**ListNode prev = null;**

**ListNode slow = head;**

**ListNode fast = head;**

**ListNode l1 = head;**

**while(fast != null && fast.next != null) {**

**prev = slow;**

**slow = slow.next;**

**fast = fast.next.next;**

**}**

**prev.next = null;**

**ListNode l2 = reverse(slow);**

**merge(l1, l2);**

**}**

**public ListNode reverse(ListNode node) {**

**ListNode prev = null;**

**ListNode current = node;**

**while(node != null) {**

**current = node.next;**

**node.next = prev;**

**prev = node;**

**node = current;**

**}**

**return prev;**

**}**

**public void merge(ListNode l1 , ListNode l2) {**

**while(l1 != null) {**

**ListNode node1 = l1.next;**

**ListNode node2 = l2.next;**

**l1.next = l2;**

**if(node1 == null) {**

**break;**

**}**

**l2.next = node1;**

**l1 = node1;**

**l2 = node2;**

**}**

**}**

**}**

Time complexity : O(n) , n is number of nodes in given linkedlist

Space Complexity : O(1) ,constant space