```
/*
// Definition for a Node.
class Node {
    public int val;
    public Node prev;
    public Node next;
    public Node child;

    public Node() {}

    public Node(int _val,Node _prev,Node _next,Node _child) {
        val = _val;
        prev = _prev;
        next = _next;
        child = _child;
    }
};
*/
class Solution {
    public Node flatten(Node head) {
        if (head == null)
            return head;
        Deque<Node> dq = new ArrayDeque<>();
        Node itr = head;
        while (itr != null) {
            if (itr.child != null) {
                if (itr.next != null) {
                    dq.offerLast(itr.next);
                }
                itr.next = itr.child;
                itr.child.prev = itr;
                itr.child = null;
            }
            if (itr.next == null && !dq.isEmpty()) {
                Node node = dq.pollLast();
                itr.next = node;
                node.prev = itr;
            }
            itr = itr.next;
        }
        return head;
    }
}
```