

#LeetCode82

<https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/description/>

Yifeng Zeng

#题目描述

Remove Duplicates from Sorted List II

#思路报告

因为是sorted linked list, 而且有重复要删除重复数字的所有node,那么意味着我们需要一个prev node来记录重复数字之前的那个node。而且第一个node就可能重复, 所以我们需要一个dummy node来辅助。那么prev从dummy开始, 用一前一后slow fast两个指针, 只要slow fast的值相等那么fast向后移动直到不等, 那么prev.next = fast就可以把所有重复的node remove掉了。如果slow fast都不相等, 那么prev slow fast同时往后移一位。

代码如下:

```
public ListNode deleteDuplicates(ListNode head) {
    if (head == null) {
        return head;
    }

    ListNode dummy = new ListNode(0);
    dummy.next = head;
    ListNode prev = dummy;
    ListNode slow = head;
    ListNode fast = head.next;
    while (fast != null) {
        while (fast != null && fast.val == slow.val) {
            fast = fast.next;
        }

        if (slow.next == fast) {
            prev = slow;
            slow = fast;
        } else if (slow.next != fast) {
            prev.next = fast;
            slow = fast;
        }
    }
}
```

```
    }  
}  
  
    return dummy.next;  
}
```

#套路总结

- 改变原list的话要借助dummy node。
- 在singly linked list里面remove掉某些node的话，需要prev。