

```

        if (left != null) {
            left.next = end;
        }
        start.next = right;
    }
}

```

删除无序单链表中值重复出现的节点

【题目】

给定一个无序单链表的头节点 `head`，删除其中值重复出现的节点。

例如：1->2->3->3->4->4->2->1->1->null，删除值重复的节点之后为 1->2->3->4->null。

请按以下要求实现两种方法。

方法 1：如果链表长度为 N ，时间复杂度达到 $O(N)$ 。

方法 2：额外空间复杂度为 $O(1)$ 。

【难度】

士 ★☆☆☆

【解答】

方法一：利用哈希表。时间复杂度为 $O(N)$ ，额外空间复杂度为 $O(N)$ 。

具体过程如下：

1. 生成一个哈希表，因为头节点是不用删除的节点，所以首先将头节点的值放入哈希表。
2. 从头节点的下一个节点开始往后遍历节点，假设当前遍历到 `cur` 节点，先检查 `cur` 的值是否在哈希表中，如果在，则说明 `cur` 节点的值是之前出现过的，就将 `cur` 节点删除，删除的方式是将最近一个没有被删除的节点 `pre` 连接到 `cur` 的下一个节点，即 `pre.next=cur.next`。如果不在，将 `cur` 节点的值加入哈希表，同时令 `pre=cur`，即更新最近一个没有被删除的节点。

方法一的具体实现请参看如下代码中的 `removeRep1` 方法。

```

public Node {
    public int value;
    public Node next;

    public Node(int data) {
        this.value = data;
    }
}

```

```

    }

    public void removeRep1(Node head) {
        if (head == null) {
            return;
        }
        HashSet<Integer> set = new HashSet<Integer>();
        Node pre = head;
        Node cur = head.next;
        set.add(head.value);
        while (cur != null) {
            if (set.contains(cur.value)) {
                pre.next = cur.next;
            } else {
                set.add(cur.value);
                pre = cur;
            }
            cur = cur.next;
        }
    }
}

```

方法二：类似选择排序的过程，时间复杂度为 $O(N^2)$ ，额外空间复杂度为 $O(1)$ 。

例如，链表 1->2->3->3->4->4->2->1->1->null。

首先是头节点，节点值为 1，往后检查所有值为 1 的节点，全部删除。链表变为：1->2->3->3->4->4->2->null。

然后是第二个节点，节点值为 2，往后检查所有值为 2 的节点，全部删除。链表变为：1->2->3->3->4->4->null。

接着是第三个节点，节点值为 3，往后检查所有值为 3 的节点，全部删除。链表变为：1->2->3->4->4->null。

最后是第四个节点，节点值为 4，往后检查所有值为 4 的节点，全部删除。链表变为：1->2->3->4->null。

删除过程结束。

方法二的具体实现请参看如下代码中的 removeRep2 方法。

```

    public void removeRep2(Node head) {
        Node cur = head;
        Node pre = null;
        Node next = null;
        while (cur != null) {
            pre = cur;
            next = cur.next;
            while (next != null) {
                if (cur.value == next.value) {
                    pre.next = next.next;
                }
                next = next.next;
            }
            cur = cur.next;
        }
    }
}

```

```

        } else {
            pre = next;
        }
        next = next.next;
    }
    cur = cur.next;
}
}

```

在单链表中删除指定值的节点

【题目】

给定一个链表的头节点 `head` 和一个整数 `num`, 请实现函数将值为 `num` 的节点全部删除。

例如, 链表为 `1->2->3->4->null`, `num=3`, 链表调整后为: `1->2->4->null`。

【难度】

士 ★☆☆☆

【解答】

方法一: 利用栈或者其他容器收集节点的方法。时间复杂度为 $O(N)$, 额外空间复杂度为 $O(N)$ 。

将值不等于 `num` 的节点用栈收集起来, 收集完成后重新连接即可。最后将栈底的节点作为新的头节点返回, 具体过程请参看如下代码中的 `removeValue1` 方法。

```

public Node removeValue1(Node head, int num) {
    Stack<Node> stack = new Stack<Node>();
    while (head != null) {
        if (head.value != num) {
            stack.push(head);
        }
        head = head.next;
    }
    while (!stack.isEmpty()) {
        stack.peek().next = head;
        head = stack.pop();
    }
    return head;
}

```

方法二: 不用任何容器而直接调整的方法。时间复杂度为 $O(N)$, 额外空间复杂度为