数组去重

```
class Solution {
public:
  int removeDuplicates(std::vector<int>& nums) {
    int n = nums.size();
    if (0 == n) {
      return 0;
    }

    int slow = 0, fast = 1;
    while (fast < n) {
      if (nums[slow] != nums[fast]) {
        nums[++slow] = nums[fast];
      }

      fast++;
    }

    return slow + 1;
  }
};
```

链表去重

```
class Solution {
public:
  ListNode* deleteDuplicates(ListNode* head) {
    if (nullptr == head) {
      return head;
    }
    ListNode *slow = head, *fast = head->next;

    while (fast) {
      if (slow->val != fast->val) {
        slow->next = fast;
        slow       = slow->next;
      }
      fast = fast->next;
    }

    slow->next = nullptr;
    return head;
  }
};
```