**1 (邓宇)**

#26

### Problem: Remove Duplicates from Sorted Array

**Description:**

Given a sorted array, remove the duplicates in place such that each element appear only *once* and return the new length.

Do not allocate extra space for another array, you must do this in place with constant memory.

For example,  
Given input array *nums* = [1,1,2],

Your function should return length = 2, with the first two elements of *nums* being 1 and 2 respectively. It doesn't matter what you leave beyond the new length.

**2 （黄思琦）**

#27

**Problem:** Remove Element

**Description:**

Given an array and a value, remove all instances of that value in place and return the new length.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

**3 （陈嘉曦）**

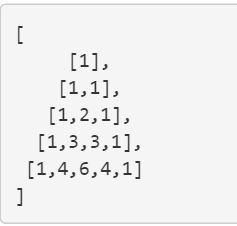
#118

### Problem: Pascal's Triangle

**Description:**

Given *numRows*, generate the first *numRows* of Pascal's triangle.

For example, given *numRows* = 5,  
Return



**4 （王开宇）**

#88

### Problem: Merge Sorted Array

**Description:**

Given two sorted integer arrays *nums1* and *nums2*, merge *nums2* into *nums1* as one sorted array.

**Note:**  
You may assume that *nums1* has enough space (size that is greater or equal to *m* + *n*) to hold additional elements from *nums2*. The number of elements initialized in *nums1* and *nums2* are *m* and *n* respectively.

**5 （梁嘉义）**

#169

**Problem:** Majority Element

**Description:**

Given an array of size *n*, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

**6 （王卓思）**

#16

### Problem: 3 Sum Closest

**Description:**

Given an array *S* of *n* integers, find three integers in *S* such that the sum is closest to a given number, target. Return the sum of the three integers. You may assume that each input would have exactly one solution.

For example, given array S = {-1 2 1 -4}, and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

**7 （杨楠）**

#11

### Problem: Container With Most Water

**Description:**

Given *n* non-negative integers *a1*, *a2*, ..., *an*, where each represents a point at coordinate (*i*, *ai*). *n* vertical lines are drawn such that the two endpoints of line *i* is at (*i*, *ai*) and (*i*, 0). Find two lines, which together with x-axis forms a container, such that the container contains the most water.

Note: You may not slant the container.

### 8 （李虹桥）

#64

### Problem: Minimum Path Sum

**Description:**

Given a *m* x *n* grid filled with non-negative numbers, find a path from top left to bottom right which *minimizes* the sum of all numbers along its path.

**Note:** You can only move either down or right at any point in time.

**9 （张晨阳）**

#78

### Problem: Subsets

### Description:

Given a set of distinct integers, *nums*, return all possible subsets.

**Note:**

* Elements in a subset must be in non-descending order.
* The solution set must not contain duplicate subsets.

For example,  
If ***nums*** = [1,2,3], a solution is:

[

[3],

[1],

[2],

[1,2,3],

[1,3],

[2,3],

[1,2],

[]

]

**10 （刘炜晨）**

#121

### Problem: Best Time to Buy and Sell Stock

**Description:**

Say you have an array for which the *i*th element is the price of a given stock on day *i*.

If you were only permitted to complete at most one transaction (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.