## 线性表

### 26. Remove Duplicates from Sorted Array

STL：

unique(Arr, Arr+n) 去除矩阵中相邻元素的重复值，返回地址

去除重复元素后的数组的个数count为count= unique(Arr, Arr+n)-Arr

distance(Arr, Arr+n) 返回两个迭代器之间的距离

### 80. Remove Duplicates from Sorted Array II

### 33. Search in Rotated Sorted Array

用二分法，O(logn)

### 81. Search in Rotated Sorted Array II

用二分法，O(logn)，对于相同的元素，直接跳过

### 4. Median of Two Sorted Arrays

用二分法，O(logn)

### 128. Longest Consecutive Sequence

用hashmap记录两端的值

### 1. Two Sum

O(n),用hashmap

### 15. 3Sum

先排序，然后左右夹逼，可以计算k个值的和

复杂度为O(max{nlogn, nk-1})

### 16. 3Sum Closest

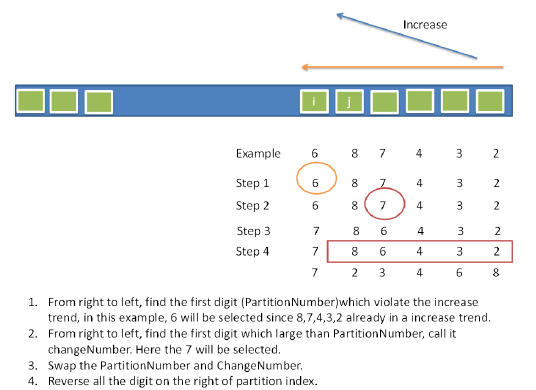
先排序，然后左右夹逼，复杂度O(n2)

### 18.4Sum

先排序，然后左右夹逼，复杂度O(n3)，可以先用一个hashmap先缓存两个数的和，最终复杂度为O(n2)

### 27. Remove Element

### 31. Next Permutation



### 60. Permutation Sequence

采用康托编码的思路，假设n个不重复的元素，第k个排列是a1,a2,….an,那么a1是第k/(n-1)!个元素，同理，k2 = k%(n-1)!， a2 = k2/(n-2)!，……

### 36. Valid Sudoku

细节实现，检查行，检查列，检查9个小方格

### 42. Trapping Rain Water

思路1：记录每个位置处的左边起的最大值和右边起的最大值，遍历数组，如果某位置处的左右最大值中的最小值大于该高度，则水量+=(min-height)，表示该位置处可以蓄水

时间O(n),空间O(n)

思路2：以最大值将数组分成两半，从左边到最大值位置，记录最大的元素，如果当前元素小于最大元素，则可以蓄水，水量+=(max-height)，时间O(n),空间O(n)

思路3：利用栈，每次将栈内小于当前高度的全部出栈，时间O(n),空间O(n)

### 48. Rotate Image

1.从外到内，一圈圈转。

2.沿着副对角线翻转一次，然后沿着水平中线转一次

3.沿着水平中线转一次，沿着主对角线翻转一次

时间复杂度O(n^2), 空间O(1)

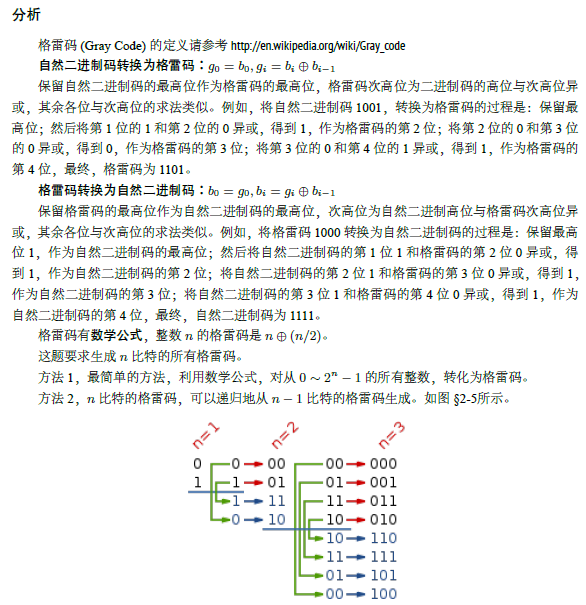
### 66. Plus One

时间O(n),空间O(1)

### 70. Climbing Stairs

斐波那契数列，时间O(n),空间O(1)

### 89. Gray Code



### 73. Set Matrix Zeroes

用两个数组记录行和列是否存在0.时间O(m\*n)，空间O(m+n)