贪心算法

1. 假设你有一个数组prices，长度为n，其中prices[i]是股票在第i天的价格，请根据这个价格数组，返回买卖股票能获得的最大收益

   (1) 你可以买入一次股票和卖出一次股票，并非每天都可以买入或卖出一次，总共只能买入和卖出一次，且买入必须在卖出的前面的某一天

   (2) 如果不能获取到任何利润，请返回0

   (3) 假设买入卖出均无手续费

示例：

```
输入：[8,9,2,5,4,7,1]
返回值：5
说明：在第3天(股票价格 = 2)的时候买入，在第6天(股票价格 = 7)的时候卖出，最大利润 = 7-2 = 5 ，不能选择在第2天买入，第3天卖出，这样就亏损7了；同时，你也不能在买入前卖出股票。
```

2. 假设你有一个数组prices，长度为n，其中prices[i]是某只股票在第i天的价格，请根据这个价格数组，返回买卖股票能获得的最大收益

   1. 你可以多次买卖该只股票，但是再次购买前必须卖出之前的股票
   2. 如果不能获取收益，请返回0
   3. 假设买入卖出均无手续费

要求：空间复杂度 O(n)，时间复杂度 O(n)

进阶：空间复杂度 O(1)，时间复杂度 O(n)

示例：

```
输入：[8,9,2,5,4,7,1]
返回值：7
说明：在第1天(股票价格=8)买入，第2天(股票价格=9)卖出，获利9-8=1
      在第3天(股票价格=2)买入，第4天(股票价格=5)卖出，获利5-2=3
      在第5天(股票价格=4)买入，第6天(股票价格=7)卖出，获利7-4=3
      总获利1+3+3=7，返回7
```

3. 在一条环路上有 n 个加油站，其中第 i 个加油站有 gas[i] 升油，假设汽车油箱容量无限，从第 i 个加油站驶往第 (i+1)%n 个加油站需要花费 cost[i] 升油。

   请问能否绕环路行驶一周，如果可以则返回出发的加油站编号，如果不能，则返回 -1。

   题目数据可以保证最多有一个答案。

示例：

```
输入：[1,2,3,4,5],[3,4,5,1,2]
返回值：3
说明：只能从下标为 3 的加油站开始完成 （即第四个加油站）
```

4. 给定一个以字符串表示的数字 num 和一个数字 k ， 从 num 中移除 k 位数字，使得剩下的数字最小。如果可以删除全部数字，则结果为 0。

   1.num仅有数字组成

   2.num是合法的数字，不含前导0

   3.删除之后的num，请去掉前导0（不算在移除次数中）

示例：

```
输入："1432219",3
返回值："1219"
说明：移除 4 3 2 后剩下 1219
```

5. 假设你是一位很棒的家长，想要给你的孩子们一些小饼干。但是，每个孩子最多只能给一块饼干。

   对每个孩子 i，都有一个胃口值 g[i]，这是能让孩子们满足胃口的饼干的最小尺寸；并且每块饼干 j，都有一个尺寸 s[j] 。如果 s[j] >= g[i]，我们可以将这个饼干 j 分配给孩子 i ，这个孩子会得到满足。你的目标是尽可能满足越多数量的孩子，并输出这个最大数值。

示例：

```
输入：g = [1,2,3], s = [1,1]
输出：1
解释：
你有三个孩子和两块小饼干，3个孩子的胃口值分别是：1,2,3。
虽然你有两块小饼干，由于他们的尺寸都是1，你只能让胃口值是1的孩子满足。
所以你应该输出1。
```

6. 给你一个整数数组 nums ， 判断这个数组中是否存在长度为 3 的递增子序列。

   如果存在这样的三元组下标 (i, j, k) 且满足 i < j < k ， 使得 nums[i] < nums[j] < nums[k] ， 返回 true ； 否则，返回 false 。

示例：

```
输入：nums = [2,1,5,0,4,6]
输出：true
解释：三元组 (3, 4, 5) 满足题意，因为 nums[3] == 0 < nums[4] == 4 < nums[5] == 6
```

7. 给定一个包含非负整数的数组 `nums` ， 返回其中可以组成三角形三条边的三元组个数。

示例：

```
输入：nums = [2,2,3,4]
输出：3
解释:有效的组合是：
    2,3,4 (使用第一个 2)
    2,3,4 (使用第二个 2)
    2,2,3
```

8. 给定一个区间的集合 intervals ， 其中 intervals[i] = [$start_i$, end_i] 。返回 需要移除区间的最小数量，使剩余区间互不重叠 。

示例：

```
输入：intervals = [[1,2],[2,3],[3,4],[1,3]]
输出：1
解释：移除 [1,3] 后，剩下的区间没有重叠。
```

回溯法

1. 给一个01矩阵，1代表是陆地，0代表海洋， 如果两个1相邻，那么这两个1属于同一个岛。我们只考虑上下左右为相邻。

   岛屿: 相邻陆地可以组成一个岛屿（相邻:上下左右） 找到最大岛屿面积， 如果没有岛屿，返回0。

   例如：

   输入

   [

   [1,1,0,0,0],

   [0,1,0,1,1],

   [0,0,0,1,1],

   [0,0,0,0,0],

   [0,0,1,1,1]

   ]

   对应的输出为3

   (注：存储的01数据其实是数字0,1)

示例

```
输入：[[1,1,0,0,0],[0,1,0,1,1],[0,0,0,1,1],[0,0,0,0,0],[0,0,1,1,1]]
返回值：4
```

2. 给出n对括号，请编写一个函数来生成所有的由n对括号组成的合法组合。

示例：

```
输入：n = 3
输出：["((()))","(()())","(())()","()(())","()()()"]
```

3. 给定一个不含重复数字的数组 `nums` ，返回其 所有可能的全排列 。你可以 按任意顺序 返回答案

示例：

```
输入：nums = [1,2,3]
输出：[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
```

4. 给定一个可包含重复数字的序列 `nums` ，按任意顺序 返回所有不重复的全排列。

   示例：

```
输入：nums = [1,1,2]
输出：
[[1,1,2],
 [1,2,1],
 [2,1,1]]
```

5. N 皇后问题是指在 n * n 的棋盘上要摆 n 个皇后， 要求：任何两个皇后不同行，不同列也不在同一条斜线上， 求给一个整数 n ， 返回 n 皇后的摆法数。

示例：

```
输入：8
返回值：92
```

6. 给你一个 无重复元素 的整数数组 nums和一个目标整数 target ， 找出nums中可以使数字和为目标数 target 的 所有 不同组合 ， 并以列表形式返回。你可以按 任意顺序 返回这些组合。

   nums中的 同一个 数字可以 无限制重复被选取 。如果至少一个数字的被选数量不同，则两种组合是不同的。

示例：

```
输入：candidates = [2,3,6,7], target = 7
输出：[[2,2,3],[7]]
解释：
2 和 3 可以形成一组候选，2 + 2 + 3 = 7 。注意 2 可以使用多次。
7 也是一个候选， 7 = 7 。
仅有这两种组合。
```