1. 给定一个未经排序的整数数组，找到最长且****连续增长的子序列****，并返回该序列的长度比如 [2,3,2,4,5,6,3] 中符合条件的是[4,5,6],程序返回数字3。如果是不要求递增，只需增长，满足的是[2,4,5,6],返回4
2. 给你一个数组 nums ,求出对应的动态和数组。数组「动态和」的计算公式为：runningSum[i] = sum(nums[0]…nums[i]).比如[1,2,3,4],返回[1,1+2,1+2+3,1+2+3+4]为[1,3,6,10]
3. 选做：小扣在秋日市集选择了一家早餐摊位，一维整型数组 staple 中记录了每种主食的价格，一维整型数组 drinks 中记录了每种饮料的价格。小扣的计划选择一份主食和一款饮料，且花费不超过 x 元。请返回小扣共有多少种购买方案
4. 两句话的不常见单词：给定两个句子 A 和 B 。 （句子是一串由空格分隔的单词。每个单词仅由小写字母组成。）如果一个单词在其中一个句子中只出现一次，在另一个句子中却没有出现，那么这个单词就是不常见的。返回由所有不常用单词构成的数组。您可以按任何顺序返回列表

如：输入：A = "this apple is sweet", B = "this apple is sour" 输出：["sweet","sour"]

1. 最大数量气球：给你一个字符串 text，你需要使用 text 中的字母来拼凑尽可能多的单词 "balloon"（气球）。字符串 text 中的每个字母最多只能被使用一次。请你返回最多可以拼凑出多少个单词 "balloon"。

**如：输入：**text = "loonbalxballpoon" **输出：**2

**输入：**text = "nlaebolko" **输出：1**

1. 旅行终点站：给你一份旅游线路图，该线路图中的旅行线路用数组 paths 表示，其中 paths[i] = [cityAi, cityBi] 表示该线路将会从 cityAi 直接前往 cityBi 。请你找出这次旅行的终点站，即没有任何可以通往其他城市的线路的城市。题目数据保证线路图会形成一条不存在循环的线路，因此只会有一个旅行终点 。

如：**输入：**paths = [["London","New York"],["New York","Lima"],["Lima","Sao Paulo"]] 输出："Sao Paulo"