1. 电面：一个中东的大哥，题目LC628 + LC42

class Solution {

public int maximumProduct(int[] nums) {

int min1 = Integer.MAX\_VALUE, min2 = Integer.MAX\_VALUE;

}

}

// 找绝对值最大的两个负数，绝对值最大的三个正数

/\*

We need not necessarily sort the given numsnums array to find the maximum product. Instead, we can only find the required 2 smallest values(min1min1 and min2min2) and the three largest values(max1, max2, max3max1,max2,max3) in the numsnums array, by iterating over the numsnums array only once.

At the end, again we can find out the larger value out of min1 \times min2 \times max1min1×min2×max1 and max1 \times max2 \times max3max1×max2×max3 to find the required maximum product.

t O(n)

s O(1)

\*/

public class Solution {

public int maximumProduct(int[] nums) {

int min1 = Integer.MAX\_VALUE, min2 = Integer.MAX\_VALUE;

int max1 = Integer.MIN\_VALUE, max2 = Integer.MIN\_VALUE, max3 = Integer.MIN\_VALUE;

for (int n: nums) {

if (n <= min1) {

min2 = min1;

min1 = n;

} else if (n <= min2) { // n lies between min1 and min2

min2 = n;

}

if (n >= max1) { // n is greater than max1, max2 and max3

max3 = max2;

max2 = max1;

max1 = n;

} else if (n >= max2) { // n lies betweeen max1 and max2

max3 = max2;

max2 = n;

} else if (n >= max3) { // n lies betwen max2 and max3

max3 = n;

}

}

return Math.max(min1 \* min2 \* max1, max1 \* max2 \* max3);

}

}

trappring rain water

Given *n* non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it is able to trap after raining.

  
The above elevation map is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped. **Thanks Marcos** for contributing this image!

**Example:**

**Input:** [0,1,0,2,1,0,1,3,2,1,2,1]

**Output:** 6

class Solution {

public int trap(int[] height) {

int leftIndex=0;

int rightIndex=height.length-1;

int max = 0;

int leftmax = 0;

int rightmax=0;

while(leftIndex<rightIndex) {

leftmax = Math.max(leftmax,height[leftIndex]);

rightmax=Math.max(rightmax,height[rightIndex]);

if (leftmax < rightmax) {

max+=(leftmax-height[leftIndex]);

leftIndex++;

} else {

max+=(rightmax-height[rightIndex]);

rightIndex--;

}

}

return max;

}

}

/\*

// Keep track of the maximum height from both forward directions backward directions, call them leftmax and rightmax.

int a=0;

int b=A.length-1;

int max=0;

int leftmax=0;

int rightmax=0;

while(a<=b){

leftmax=Math.max(leftmax,A[a]);

rightmax=Math.max(rightmax,A[b]);

if(leftmax<rightmax){

max+=(leftmax-A[a]); // leftmax is smaller than rightmax, so the (leftmax-A[a]) water can be stored

a++;

}

else{

max+=(rightmax-A[b]);

b--;

}

}

return max;

\*/

1. 1. sum of two fractions，输出为最简格式  
   2. Prime Factorization，返回ArrayList<Integer>
2. 2面一共问了两道题，总体还是很简单的，第一道算法，第二道系统设计  
     
   第一道算法题比较简单，题目是问给一个二叉树，找到一个二叉树中最短的路径。我用的是递归，然后维护一个全局的变量来存储最短路径。  
     
   第二道系统设计是设计一个类似于dropbox的系统，能够让用户上传，浏览，下载文件。以下是我的设计  
     
   先讨论了一下系统的throughput和output，做了一些假设，然后做了一个简单的系统的设计  
   client => front-end server => load balancer => back-end => load balancer => file system server  
     
   完了之后面试官问，如果用户上传一个5gb的文件，系统应该怎么设计？然后我就说为了性能的考虑，将文件在前端进‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍行拆分，然后在后端重组。这个可以灵活回答。  
     
   希望对大家有帮助！ 求积分。
3. 形式是一边打电话（会提供电话号和conference code）， 一边在一个叫coderpad的在线面试工具里写码。时长1小时，共两道题。  
   因为需要run code，所以最基本的要求是你的code能跑。。。  
   面试官会自己设置tesst cases 当你的code通过了以后 你可以自己补充一些test cases[hide=150]  
   第一题是在字符串里找找最长的substring并以tuple形式输出首尾idx

<https://leetcode.com/problems/longest-substring-without-repeating-characters/solution/>

brute force

public class Solution {

public int lengthOfLongestSubstring(String s) {

int n = s.length();

int ans = 0;

for (int i = 0; i < n; i++)

for (int j = i + 1; j <= n; j++)

if (allUnique(s, i, j)) ans = Math.max(ans, j - i);

return ans;

}

public boolean allUnique(String s, int start, int end) {

Set<Character> set = new HashSet<>();

for (int i = start; i < end; i++) {

Character ch = s.charAt(i);

if (set.contains(ch)) return false;

set.add(ch);

}

return true;

}

}

sliding window

public class Solution {

public int lengthOfLongestSubstring(String s) {

int n = s.length(), ans = 0;

Map<Character, Integer> map = new HashMap<>(); // current index of character

// try to extend the range [i, j]

for (int j = 0, i = 0; j < n; j++) {

if (map.containsKey(s.charAt(j))) {

i = Math.max(map.get(s.charAt(j)), i);

}

ans = Math.max(ans, j - i + 1);

map.put(s.charAt(j), j + 1);

}

return ans;

}

}  
第二题是找grid里的max sum path，起点是左下角，终点是右下角。我起初用的是一个matrix dp table，后来面试官让我optimize一下，就用了arr table  
[/hi‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍de]

<https://leetcode.com/problems/minimum-path-sum/>

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.

**Example:**

**Input:**

[

  [1,3,1],

[1,5,1],

[4,2,1]

]

**Output:** 7

**Explanation:** Because the path 1→3→1→1→1 minimizes the sum.

public class Solution {

public int minPathSum(int[][] grid) {

int[][] dp = new int[grid.length][grid[0].length];

for (int i = grid.length - 1; i >= 0; i--) {

for (int j = grid[0].length - 1; j >= 0; j--) {

if(i == grid.length - 1 && j != grid[0].length - 1)

dp[i][j] = grid[i][j] + dp[i][j + 1];

else if(j == grid[0].length - 1 && i != grid.length - 1)

dp[i][j] = grid[i][j] + dp[i + 1][j];

else if(j != grid[0].length - 1 && i != grid.length - 1)

dp[i][j] = grid[i][j] + Math.min(dp[i + 1][j], dp[i][j + 1]);

else

dp[i][j] = grid[i][j];

}

}

return dp[0][0];

}

}

**Complexity Analysis**

* Time complexity : O(mn)*O*(*mn*). We traverse the entire matrix once.
* Space complexity : O(mn)*O*(*mn*). Another matrix of the same size is used.

public class Solution {

public int minPathSum(int[][] grid) {

int[] dp = new int[grid[0].length];

for (int i = grid.length - 1; i >= 0; i--) {

for (int j = grid[0].length - 1; j >= 0; j--) {

if(i == grid.length - 1 && j != grid[0].length - 1)

dp[j] = grid[i][j] + dp[j + 1];

else if(j == grid[0].length - 1 && i != grid.length - 1)

dp[j] = grid[i][j] + dp[j];

else if(j != grid[0].length - 1 && i != grid.length - 1)

dp[j] = grid[i][j] + Math.min(dp[j], dp[j + 1]);

else

dp[j] = grid[i][j];

}

}

return dp[0];

}

}

1. 分数以array形式表现，比如 [[1, 2], [1, 3],[5, 6]], 就是要写个函数能计算1/2 + 1/3 = 5/6。解法参考 <https://www.geeksforgeeks.org/program-to-add-two-fractions/>  
   第二题很简单，reverse string，如果用python非常简单return str[::-1]

class Solution {

public String reverseString(String s) {

// return new StringBuilder(s).reverse().toString();

char[] charArr = s.toCharArray();

int i = 0;

int j = s.length() - 1;

while (i < j) {

char temp = charArr[i];

charArr[i] = charArr[j];

charArr[j] = temp;

i++;

j--;

}

return String.valueOf(charArr);

}

}  
第三题有一个array的字母，每个字母都有weight，比如 [{a:3}, {b:9}, {c:11}, {d:2}...]，如何用最少的字母去代表一个数字。这个就是coin sum的问题，解法参考 [https://www.geeksforgeeks.org/fi ... ‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍that-make-a-change/](https://www.geeksforgeeks.org/find-minimum-number-of-coins-that-make-a-change/)

1. Goldman 两轮电面  
     
   第一轮 2 道题  
     
   第一题是那个  Find Minimum in Rotated Sorted Array  
     
   第二题是计算 Average grades 在 glassdoor 上有

第二轮 1 道题  
  
rain water trap   
  
两轮都是senior manager， 第一轮是印度人，第二轮是东欧的。  
  
还问了 java8 里面的东西，比如interface default method。  
  
还有关于map 有哪‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍几个 thread safe 的，concurent hashmap 和 sychonized hashmap。  
  
总体来讲和software 难度一样。

1. 1. **is power of 10**, 给一个整数，返回是否是power of ten。  
   我一开始有点紧张，蒙了，以为是问10的倍数，后来又反应了一下，用转换成字符串的方法做的。10的倍数，又是整数，那只有首位为1剩余为零的字符串满足条件，<=0的整数直接返回false。  
     
   做完以后跑了几个test case，又自己写了几个case，过了。  
     
   2. [刷题](http://www.1point3acres.com/bbs/forum-84-1.html)网站-而林久。该公司的高频题，之前刚刷过，顺利过了。
2. codepad两题，java  
   1."aaabbc" -> "a3b2c1"  
   2.给一个hashset比如abl,bla,ab,ssssss， 给一个string，比如angelababy。找出那个set里能用angelababy拼成的词的长度最长的词:abl & bla。  
   利用第一问brute force肯定是能做的，但是怎么优化呢。  
   说实话这题我现在也不会，没听懂面试官给的提示，要用什么数据结构去写啊？

发一个热乎的小金人店面，国人面试官，题目比较正常，不像大互联网公司。三道题45分钟，两道easy，一道medium。（1）10的方次， 不过多问了一些变形，比如输入是double； （2）质数分解‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍； （3）两列数组求中值。

1. 第一個是力叩肆伍柒變形 回傳圈的大小  
   ‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍  
   第二個是力叩貳零玖
2. 两道题，都是经典面经题：1. atoi，2.highest average。关于写test case， 不用太担心，题目里面已经给你一个test class，只需要多加几个test ca‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍se就可以。
3. 1. second smallest integer  
   2. debug, given a large document as a string, find the shortest distance between two given words.  
   如果找不到，return 0；  
   比如 docuement = "ab cd ef", 求 ab， cd的最短中间‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍距离就是 a与b 之间，加上一个空格 加上 c与d之间也就是 1+ 1+ 1 = 3；  
   code是给好的，test case也是，但是有很多错误。  
   要注意顺序可能是反的，string可能多种字符，可能多个空格，edge case 很多。建议和面试官一点一点讨论。

|  |
| --- |
| 1. Implement BST的put和contains methods以及改正in order travasal的code。做的有点慢，刚写完一小时就没了，可能原来人家准备了两题 |

1. 一个听不出口音的大叔？大哥？  
   先互相自我介绍了一下 然后就coderpad  
   1. find the first unique char  
   2.求最高average。自己写了string转int的方程 但是面试官说可以直接用parseInt  
   问了半天怎么写test case unit test之类的 因为不太会unit test 过程稍微尴尬了一下  
   别的基本没问什么特别的问题  
   第一题的时候 代码里没import java 导致hashmap不识别 额呵 找了半天没找到问题在哪 第二题学精了 先import了java 结果面试官说 这道题代码‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍里有 也是非常尴尬了
2. 分组anagram：就是给一堆string，要把他们分组，每组都是anagram  
   两个分数相加  
   sum最大的sub array问题。其实这题很久之前自己写过一遍，但当时面试现场出了bug也没有bug free
3. 第一道题 high\_avarage score   
   第二道题 find "a" in "a aa aab bac" return inde‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍x 地里有
4. 1, 介绍5-10分钟，告诉了他们组干嘛的，要做什么东西，然后就是问我背景，我也简单的说了  
     
   2，sliding window， 给一个str，找出里面最长的联系重复子串，不算太难，边界条件卡了我一会，然后就是写一些额外的test cases，然后说下为什么要写这些case  
     
   3， 给了一些数据，每条里面是一个名字一个分数，然后问这些数据里面最高的平均分是多少，就是同名分数加起来除一下N，然后比较就可以了，然后就是自己写test c‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍ases，他说要注意负数和rounding吧，我都写了test cases，不知道对面满不满意

<https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=523564&extra=page%3D1%26filter%3Dsortid%26sortid%3D311%26searchoption%5B3089%5D%5Bvalue%5D%5B2%5D%3D2%26searchoption%5B3089%5D%5Btype%5D%3Dcheckbox%26searchoption%5B3046%5D%5Bvalue%5D%3D58%26searchoption%5B3046%5D%5Btype%5D%3Dradio%26sortid%3D311%26orderby%3Ddateline>

1. 刚刚结束的高盛面试，面试官一开始先聊了聊简历，问了问我工作的内容和学校背景，然后问了我最喜欢的课程，当时我说了data structure，于是面试官很随性地说，那我们就实现hashmap吧。看来聊天内容还是很能决定面试官的问题的。我先讲述了一下hashmap的实现机制，然后说了colloion的两种stragety之后，直接给了最经典的array+linkedlist的解法，然后写了几个test cases，高盛的面试官感觉还挺看中test cases的。  
   第二题考的是 3 sum，还是比较标准的，我先讲述了一下暴力破解的问题，然后就说可以先对数组进行排序，完了使用two pointers，因为这样可以把时间缩减到O(n^2)，然后跑了几个test cases，面试官最后就让我问他问题，问了一些公司work load life balanc‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍e的问题，面试就结束了。
2. 1. find min in a rotated sorted arra  
   2. 里扣珥陵酒 这个地里提过很多遍了
3. 1. implement HashMap  
   ‍‌‌‌‍‍‍‍‍‍‌‌‌‌‌‌‌‍2. prime factorization
4. 1. 给两个sorted arrays找出median。  
   2. Find prefix in a document. 给一个很长的document和一个prefix，要求在document里面找出prefix所在的位置，不分大小写。如document="a aa Abc bacd bce"，prefix="a"，return=[0, 2, 5]。  
     
   题目挺简单，但是他们家追求bug free，debug还是挺麻烦的。