

Linux_17 级算法题

summer 第三周

1. 给定一个无序数组，包含正数、负数和 0，要求从中找出 3 个数的乘积，使得乘积最大，要求时间复杂度： $O(n)$ ，空间复杂度： $O(1)$

（题目来源：拼多多面试题）

输入描述：

无序整数数组 $A[n]$

输出描述：

满足条件的最大乘积

示例：

输入：

3 4 1 2

输出：

24

2. 有两个用字符串表示的非常大的大整数, 算出他们的乘积, 也是用字符串表示。不能用系统自带的大整数类型。（题目来源：拼多多面试题）

输入描述：

空格分隔的两个字符串，代表输入的两个大整数

输出描述：

输入的乘积，用字符串表示

示例：

输入：

72106547548473106236 982161082972751393

输出：

70820244829634538040848656466105986748

3. 六一儿童节，老师带了很多好吃的巧克力到幼儿园。每块巧克力 j 的重量为 $w[j]$ ，对于每个小朋友 i ，当他分到的巧克力大小达到 $h[i]$ (即 $w[j] \geq h[i]$)，他才会上去表演节目。老师的目标是将巧克力分发给孩子们，使得最多的小孩上台表演。可以保证每个 $w[i] > 0$ 且不能将多块巧克力分给一个孩子或将一块分给多个孩子。（题目来源：拼多多面试题）

输入描述：

第一行： n ，表示 h 数组元素个数

第二行： n 个 h 数组元素

第三行： m ，表示 w 数组元素个数

第四行： m 个 w 数组元素

输出描述：

上台表演学生人数

示例：

输入：

3

2 2 3

2

3 1

输出：

1

4. 假设一个探险家被困在了地底的迷宫之中，要从当前位置开始找到一条通往迷宫出口的路径。迷宫可以用一个二维矩阵组成，有的部分是墙，有的部分是路。迷宫之中有的路上还有门，每扇门都在迷宫的某个地方有与之匹配的钥匙，只有先拿到钥匙才能打开门。请设计一个算法，帮助探险家找到脱困的最短路径。如前所述，迷宫是通过一个二维矩阵表示的，每个元素的值的含义如下 0-墙，1-路，2-探险家的起始位置，3-迷宫的出口，大写字母-门，小写字母-对应大写字母所代表的门的钥匙。（题目来源：拼多多面试题）

输入描述：

迷宫的地图，用二维矩阵表示。第一行是表示矩阵的行数和列数 M 和 N 后面的 M 行是矩阵的数据，每一行对应与矩阵的一行（中间没有空格）。M 和 N 都不超过 100，门不超过 10 扇。

输出描述：

路径的长度，是一个整数

示例：

输入：

5 5

02111

01a0A

01003

01001

01111

输出：

7

5. 在商城的某个位置有一个商品列表，该列表是由 L1、L2 两个子列表拼接而成。当用户浏览并翻页时，需要从列表 L1、L2 中获取商品进行展示。展示规则如下：

1. 用户可以进行多次翻页，用 `offset` 表示用户在之前页面已经浏览的商品数量，比如 `offset` 为 4，表示用户已经看了 4 个商品
2. `n` 表示当前页面需要展示的商品数量
3. 展示商品时首先使用列表 L1，如果列表 L1 长度不够，再从列表 L2 中选取商品
4. 从列表 L2 中补全商品时，也可能存在数量不足的情况

请根据上述规则，计算列表 L1 和 L2 中哪些商品在当前页面被展示了

（题目来源：拼多多面试题）

输入描述：

每个测试输入包含 1 个测试用例，包含四个整数，分别表示偏移量 `offset`、元素数量 `n`，列表 L1 的长度 `l1`，列表 L2 的长度 `l2`。

输出描述：

在一行内输出四个整数分别表示 L1 和 L2 的区间 `start1`，`end1`，`start2`，`end2`，每个数字之间有一个空格。 注意，区间段使用半开半闭区间表示，即包含起点，不包含终点。如果某个列表的区间为空，

使用 $[0, 0)$ 表示，如果某个列表被跳过，使用 $[len, len)$ 表示， len 表示列表的长度。

示例：

输入：

2 4 4 4

1 2 4 4

4 1 3 3

输出：

2 4 0 2

1 3 0 0

3 3 1 2

6. 有 n 只小熊，他们有着各不相同的战斗力。每次他们吃糖时，会按照战斗力来排，战斗力高的小熊拥有优先选择权。前面的小熊吃饱了，后面的小熊才能吃。每只小熊有一个饥饿值，每次进食的时候，小熊们会选择最大的能填饱自己当前饥饿值的那颗糖来吃，可能吃完没饱会重复上述过程，但不会选择吃撑。

现在给出 n 只小熊的战斗力和饥饿值，并且给出 m 颗糖能填饱的饥饿值。

求所有小熊进食完之后，每只小熊剩余的饥饿值。（题目来源：拼多多面试题）

输入描述:

第一行两个正整数 n 和 m , 分别表示小熊数量和糖的数量。($n \leq 10$, $m \leq 100$) 第二行 m 个正整数, 每个表示着颗糖能填充的饥饿值。接下来的 n 行, 每行 2 个正整数, 分别代表每只小熊的战斗力和当前饥饿值。 题目中所有输入的数值小于等于 100。

输出描述:

输出 n 行, 每行一个整数, 代表每只小熊剩余的饥饿值。

示例:

输入:

2 5

5 6 10 20 30

4 34

3 35

输出:

4

0

例子说明:

第一只小熊吃了第 5 颗糖

第二只小熊吃了第 4 颗糖

第二只小熊吃了第 3 颗糖

第二只小熊吃了第 1 颗糖