

UBER OA (2020.12 - 2021.2)

Q1

- canMakeTriangle, 给一个包含正数的无序数组, 问从0 到n-2 有多少个边的组合能形成三角形, 比如i = 0, 取的是 arr[0], arr[1] 和 arr[2]三个边, 能组成三角形的条件是任意两个边的加和比大于第三条边。简单one pass就过
- 给定一个数组A 按条件输出另一个数组B(长度len(A) - 2) 如果A中连续的三个数单调 则在B中输出1否则输出0
- 给一个字符串, 数连续三个字符都不相同的个数, 如aaaaaa:0个, abcdef:4个
{abc}{bcd}{cde}{def}: 这道题写一个子函数判断一个长度为3的字符串, 是否里面的字符两两不相同, 然后对原字符串采用类似moving window暴力求解即可

Q2

- 给一个 像 rotated sorted 的数组, 比如 [4,5,1,2,3], 问能不能左右shift number 转换成 1 to n 的 升序consecutive array 或者是 n to 1的降序consecutive array, test case 里有些是false的, 比如[4,2,3,1]就不能
- 分割字符串为3部分 a b c s.t. s = a + b + c并且a + b, b + c和c + a互不相同 求满足条件的分割的数量
- 对于一个string, 找prefix里最长的palindrome (并且长度大于1), 删掉。并且重复上述过程, 把最后得到的string输出
- 给两个纯数字的字符串, 如"987"和"654", 然后从最低位开始两两求和, 输出最终的字符串, 两个输入字符串都没有leading zeros。如"987"和"654"输出"151311", 我是先把字符串转成int数组, 然后安位求和, 最后再转换回string。

Q3

- 是新闻排版, input an array of lines, each line is an array of words. (array of array of string) 然后给你一个屏幕宽度, 每一行贪心地放 word, 直到排满然后换行继续排, 剩余的地方用空格去 padding, 保证每一格都要凑够到屏幕宽度。没啥难的, 就是细节比较多 (每一行前后都要加一个星号, 最前面和最后面要加一排星号), 写起来倒很快。

比如给你 `[["hello", "world"], ["I", "love", "cats", "and", "dogs"]]`, 宽度 12, 第一行左对齐第二行右对齐, 要求输出

```
*****
*hello world *
* I love cats*
*   and dogs*
*****
```

- 给定 $n \times n$ 的方阵, 要求剥洋葱似的一层层分别排序重新填回这个方阵。比如说

```
3 2 4      1 2 3
1 9 6  ==> 8 9 4
5 7 8      7 6 5
```

- 给定一个字符方阵 从左下角到右上角的对角线依次编号 为这几个对角线字符串排序
- 地里原题, 方形的matrix旋转K次90度, 求最后的矩阵什么样, 对角线的元素跳过, 比如 `[0][0], [1][1]` 这些点是不参与旋转的
- Memory allocate, 一个数组a代表内存, 值为0表示空闲, 1表示已分配
如果是allocate, 找到第一个连续的长度为X的空闲内存, 将其设置为1, 并且记录这段内存的id (id是auto increments的)
如果是erase, 会给出id, 找到id对应的内存段, 将其全部设置为0。并且记录这段内存的长度
会给一系列allocate和erase操作, 输出每一次操作的记录值
- 给一个二维矩阵, 每一行的第一列的数字, 称之为pivot, 要求对这些pivot进行排序。
排序的comparator不是基于这些pivot的值, 而是他们的“右上-右下-对角线sum”, 这个对角线sum的定义是, 从这个pivot 开始向右上方挪动, 到顶之后再像右下方挪动直到到达右边界或者下边界, 这样遍历下来的sum称之为对角线sum。这道题, 我是写一个子程序计算给定位置的pivot值, 然后返回一个tuple(pivot,val)这里val是原始位置的值 (因为这个是最后要输出的), 然后对第一列的每一行元素计算pivot, 进行排序即可。

- 二维数组 $m \times m$, 一系列繁琐诡异的操作: 取里面所有 $n \times n$ 的小数组, 循环所有小数组, 求小数组元素和。找到最大元素和。输出构成最大元素和的所有distinct小数组的元素。

这题input是个二维数组, 这个二维数组在图上面展示出来类似于黑白棋, 最左上角的点是黑色的, 与其相邻的点是白色, 以此类推黑色和白色相互交错

然后又给了数组类似于 `[[0,1,2],[0,2,3]]`, 意思是其中 `[0,1,2]` 表示最左上角以 (0, 1) 为起点的 2×2 的矩阵, 矩阵中的黑白棋子按照各自的颜色分组进行排序,

进行了两次排序后输出得到的数组

eg : input:

[[1,4,3,2],[8,4,7,1],[1,5,2,1]]

[[0,1,3]]

output: [[1,1,1,2],[8,2,4,3],[1,5,4,7]]

Q4

- 给一个包含正负数的数组，求所有pair的sum是一个perfect square的数量。

比如 [-1, 0, 1, 18, 3]

pair里的i, j 的条件是 $i \leq j$, 就是说同一个数字能够重复使用, 在这个例子里 最大的pair sum就是36 (18 +18), 那么有可能出现的perfect square就有0, 1, 4, 9, 16, 25, 36

那么 [-1, 1], [0,1],[1,3], [18,18]就是所有的pair

输出为4.(应该是5 还有[0,0])

- 我用了一个hashmap的解法，不知道为什么最后一个test case超时

第四是原本有一个 N 个数组成的数组，但这个数组不会给你，只会给你这个数组里两两组成的 pair，一共有 N-1 个。这些 pair 的顺序是 shuffled，并且每个 pair 里的两个数也可能会互换顺序。要求还原数组。

比如给你的是 [[5, 4], [2, 4], [1, 3], [1, 5]], 要求输出 [3, 1, 5, 4, 2] (或者 [2, 4, 5, 1, 3] 也是正确答案)

这个题恶心的地方在于，如果用 recursive 的 dfs，它就会有几个 case 过不了，你还不知道为啥过不了。我 debug 了半个小时，各种地方都试了，还是过不了。

最后灵机一动怀疑可能是 recursion 次数超了（毕竟题里说 N 最大有 10^6 ），改成了个 iteration 的才过。

- 给定整数数组A和整数x，找出所有 (i, j) 的个数，要求满足 $i \neq j$, $A.concat(A[j]) == x$ 或者

$A[j].concat(A) == x$, 如果这俩都满足就算两次。举个栗子：

$A = [12, 121, 2, 12]$, $x = 1212$

那么 12, 12 \Rightarrow 1212, 这个算两次，因为index (0, 3) 和 (3, 0) 都满足

121, 2 \Rightarrow 1212

所以最终输出3。

- 给定一个整形数组 求出所有满足条件的子数组的数量 条件为子数组中的数字至少有k个不同的数字只出现一次
- 地里的原题，给你一个list，给你另外一个list of list，问你从后面这个list of list里能否拼出前面这个list，比如 [3,2,5,1,4] [[5,1][3,2], [4]]就可以拼出来

这我觉得corner case主要是要么拼的时候字母对不上，要么都拼进去了长度不对，两种都考虑下应该就没问题了

- arr数组，找到满足如下条件的pair数目
 $i \leq j \text{ arr}[i] + \text{rev}(\text{arr}[j]) == \text{arr}[j] + \text{rev}(\text{arr}[i])$
 $\text{rev}(\text{arr}[i])$ 表示将一个数反过来，比如123 -> 321, 20 -> 2
- 提供一个数组，数一下两两互为数字变形体的个数，数字变形体的定义是，如果一个数把它每一个数字位置改变就能和另一个数相同，它们就是数字变形体比如"123"和"321"和"312"都是数字变形体。这道题我一开始写了个子函数，判断两个数字是否是变形体。然后暴力两层循环数一共有几对，通过了基本的测试，但是隐藏测试超时。后来，对每一个数进行编码，比如 $123 = 1*100 + 1*10 + 1$, $1123 = 2*100 + 1*10 + 1$ ，类似counting sort的思想，这样直接比较两个数encoding后是否一样如果一样就是变形体，结果时间复杂度 $O(n^2)$ 还是过不了所有的隐藏测试，最后采用类似hash map的思路，建一个count字典，记录每个编码到目前为止出现的次数，遍历整个数组， $\text{sum} += \text{count}[\text{val}]$ ，最后返回sum。达到 $O(n)$ 时间复杂度，通过了所有的测试。这道题类似力扣死侍酒。
- subarray滑动窗口问题，求最长的subarray，里面相邻的元素差小于k
- 第四题，给一个数组，和一个k，找有多少个连续且长度 $\geq k$ 的subarray里面没有重复的数字

比如 {1,2,1,1} k = 2那么返回就是2，因为subarray是1, 2和2, 1

如果是这个{1,2,3,4,1} k = 3, 返回值就是4

subarray: {1,2,3} {1,2,3,4} {2,3,4} {2,3,4,1}

- 两个unsorted array, a 和 b 找 lower bound $\leq a[i] * a[i] + b[j] * b[j] \leq \text{upper bound}$ 。返回符合这个条件的一个有多少个