Editorial For Legilimens Contest 1

Chen, Ye , Liu

2018 年 3 月 17 日

A Nearly Lucky Number

题意

判定一个数里'4' 和'7' 的数量是不是一个只由'4' 和'7' 组成的数字。

题解

直接以字符串读入该数字，扫描每一位的数字，若是'4' 或'7'，给计数器加一。最后看看计数器的数字是否也符合该条件。实际上因为位数不超过 19，那么计数器的数字不超过 19，那么符合条件的幸运数字只有 4 和 7，直接判断计数器数字是否等于 4 和 7 即可。

B Foe Pairs

题意

给出一个 n 的全排列，指定一些不能出现在同一区间内的数对，求合法的区间数量。

题解

考虑相邻的两个区间左端点，显然对应的最远右端点递增，于是可以用首尾两个指针来维护每个左端点对应的最远右端点即可得到答案。加入（删除）一个新数字时，将数字在区间里对应的"非法数字"数量加入计数器（从计数器减去）即可。时间复杂度为 $O(n)$。

C Pluses everywhere

题意

给定一个长度为 n 的数字，在数字间添 k 个'+'，使得其成为一个合法的式子，求所有合法式子答案的和。

题解

对于当前第 i 位来说，该位若在个位上出现，那么第 i 位和第 $i+1$ 位中间肯定有一个"+"，剩下的 $k-1$ 个"+"分布在剩下的 $n-2$ 个空隙中，所以出现的总次数是 $C(n-2,k)$。同理，在十位上出现的总次数是 $C(n-3,k)$。于是每个数字的贡献值就可以求出来了，累加即可。

所以大体思路是遍历所有可能出现的位数，从个位开始，分成两部分计算，一部分用前缀和计算出前面所有的在该位上的贡献和，另一部分算出当前位置在该位上的贡献值。

对于求组合数，可以先将阶乘预处理出来，然后用乘法逆元求出组合数的值：

$$C(n,m)=\frac{n!}{m!*(n-m)!}$$

再细致分析，我们不妨从末尾到首开始看每一位对总和的贡献：

倒数第一位：贡献了 $C(n-1,k)$ 次个位数

倒数第二位：贡献了 $C(n-2,k-1)$ 次个位数，$C(n-2,k)$ 次十位数

倒数第三位：贡献了 $C(n-2,k-1)$ 次个位数，$C(n-3,k-1)$ 次十位数，$C(n-3,k)$ 次

倒数第四位：贡献了 $C(n-2,k-1)$ 次个位数，$C(n-3,k-1)$ 次十位数，$C(n-4,k-1)$ 次百位数，$C(n-4,k)$ 次千位数

其实每位数贡献很明显，不难想到预处理出 $C(x,k-1)$ 和 $C(x,k)$ 然后利用前缀和思想统计即可。

扩展：请自行思考如果题目变成"至多 K 个"怎么做。

D Youngling Touranament

题意

对于一个序列，定义答案为排序之后，某个位置不小于之前的前缀和的位置个数。给定一些修改，求每次修改之后的答案。

题解

假设存在一个位置 i 满足条件，则显然下一个满足条件的数字不小于 [1..i] 数字和，即不小于 a[i]+[1..i-1] 数字和，于是满足条件的 $a[i]$ 增长速度接近斐波那契数列，

于是答案为 $log(MAXa[i])$ 级别。于是每一次使用 set 寻找可能满足条件的下一个数字，用树状数组或者线段树维护前缀和检验。复杂度为 $O(nlog(n)*log(MAXa[i]))$

E Unique Party

题意

题意比较啰嗦，等价于找到一个最大的子矩阵满足子矩阵的中位数大于等于 h（若有偶数个数字，取第 $\frac{n}{2}$ 大的数字为中位数）。

题解

对于矩阵中每个数字，若大于等于 h，则置为 1，否则 -1。问题转化为找最大的矩阵和大于等于 0 的子矩阵。

枚举上下边界,得到前缀和 s,问题转化为找到满足 $i<j, s[i]\leq s[j]$ 的 $max(j-i)$，对于 $s[i]$，显然当 i 递增时 $s[i]$ 递减，否则显然不优，$s[j]$ 同理。处理出两个单调的数组，各一个指针统计答案即可。

F Almost Union-Find

题意

请维护以下 3 个操作:

操作 1，将两个元素所在集合合并。

操作 2，将一个点加入另一个元素所在集合。

操作 3，询问一个点所在集合的元素个数和元素和。

题解

对于操作 2，建立新点，把原来的点的贡献去掉。其他的操作是基本的并查集操作。

G Handling a Spaceship

题意

交互，N 维空间内 N 个线性无关的向量 A_i 和数组 $C[N][M]$，保证 C 每行单调上升且有一位置为零，每次查询 $P[N](1<=P[i]<=M)$，返回向量 $\sum C[i][P[i]]*A_i$，要求 120 次内得到零向量,$N,M\leq 100$.

题解

用 $N+1$ 次得到 A_i 的单位向量，用高斯消元求得逆矩阵，随后每次查询时就能 $O(N^2)$ 得到每个 $C[i]$ 的 sgn，一同二分即可。

H Tree Destruction

题意

给出一棵树，每一次选择两个叶子节点，将两点间路径长度加入答案，并选择一个点删掉。重复操作直到只剩下一个点。求答案的最大值。

题解

我们考虑这样一种删除方法：每一次选择一个不在直径上的点，选择和直径两端较远的那一端做操作，然后删除这个点。最后只剩下直径时答案是确定的。

先考虑我们找树的直径的方法，从任意一个点 dfs，找到距离最远的点 A，然后再 dfs 一次找到对于点 A 最远的点 B, 点 A 到点 B 就是该树的一条直径。该方法的正确性是由于对于树上任意一点，离其距离最远的点必是某条直径的一端，正确性可以由反证法证明。

对于树上任意一点，它能够对答案作出的最大贡献是到离其最远的点，由上述可知该点必为树某条直径的一端。所以按照我们的删除方法，每个不在直径上的点的贡献都是它能够做到的最大值。对于直径上的点，因为树的直径长度唯一，倘若存在更优的方法使其作出更大的贡献，那必然是取另一条直径的某端，则该点变为“不在直径上的点”，若只考虑取直径上的点为另一端，那么显然对于直径上的所有点，由外而内一步步收缩的贡献值最大。

综上，证明了该贪心策略的正确性，实际上通过反证法应该可以更直观地证明，请自行思考。

I Crazy Town

题意

平面被一些直线分成了一些区域，给定两个点，问两点之间至少需要经过多少个区域。

题解

易证，如果询问的两点分别在某直线分割的不同区域，那么这条线肯定需要被跨越。所以我们需要做的就是跨越这些直线，答案就是这样的直线条数。

注意到直线是以一般式 $Ax+By+C=0$ 给出，我们检验只需要将两点直接代入 $Ax+By+C$，然后看看符号是否不同即可。时间复杂度为 $O(n)$。

J Consanguine Calculations

题意

给出两个父母和一个孩子的血型（其中一个未知），问未知项的所有可能血型。

题解

枚举血型，然后检查这种亲子血型关系是否合法。注意只有一种可能时不需要大括号。

K String Modification

题意

给定串 s 和 t，问能否通过操作将 s 串变为 t 串。每次操作可选择 s 中一个字母，在其后面添加一个不同的字母。

题解

首先，t 串中必须存在一个子序列等于 s 串，这是必要条件 1。

然后再考虑对于 s 串中字母 c，在 t 串中若要补充为 c+str，无论 str 是什么串，总可以生成，除非 str 里字母全都为 c。此时可以考虑将 str 里的字母留给 s 串中前一个字母来生成，由此，所以对于 s 串开头第一个字母，我们需要特殊考虑，易得 s 和 t 两个串的开头都必须为同一个字母 c，我们定义 A 为 s 串开头连续 c 的个数，B 为 t 串开头连续 c 的个数，则 A 必须大于等于 B 才能生成 (不需要生成更多的 c，且可生成 t[B+1] 来将其分隔开变为一般情况)。