

codeforces round 659(div2)

=====入坑以来第一次爆零，我裂开了.....

A. Common Prefixes

题意：给 n 个数字，要求输出 $n+1$ 个字符串，满足，第 $i+1$ 个字符串和第 i 个字符串有 a_i 个相同的地方。

题解：每次前 a_i 个字符照上面复制下来，后面所有数，就填上面所有字符的后一位即可。

B Koa and the Beach (Easy Version)&&(hard Version)

题意：一个人要从0号沙滩到 $n+1$ 号岛屿，中间有 n 片海，每片海都有一个深度 d_i 还存在潮汐的作用，潮汐在呈 $[0,1,2,\dots,k-1,k,k-1,\dots,1]$ 的规律按照时间循环变化，在一个时刻海域的深度为潮汐高度加深度，有一个安全深度 s 而且他移动一次需要时间 t ，它可以在任何一个海域停留任何一个时间。问他能否安全的到达目标岛屿？

题解1：对于简单的版本，考虑使用 dp 的思想，用 $dp[i][j]$ 表示在时间为 j 时在第 i 个海域的可能性，转移时考虑 $dp[i][j-1]$ 与 $dp[i-1][j-1]$ 有没有存在一个1，有就可以转移，时间总量可以开大一点，开 $2*k*n$ 最后遍历 $dp[n][j]$ 寻找是否存在 1 即可，至于判断单个状态是否成立，这很简单。

题解2：上面那种比较暴力得做法，明显过不了大样例，所以要想一个 $O(n)$ 的想法，其实就是设计一个无论在那种情况都能占优的策略，可以发现，在降潮的时候走是有优势的，如果这时候到 $i+1$ 会有危险，那就等到 $i+1$ 没有危险时再走，二长时间呆在原地，潮水只会往下跌，所以只会越来越“安全”，所以只要找可以等到涨潮的点即可，即满足 $k+d_i \leq s$ 的点，在这个点等到潮水涨到最高，然后再开始继续走，当然，一开始要判断不成立，两种情况，一种是已经涨潮，而对面已经过不去了，还有就是存在一片海域 $d_i > s$ ，那么也一定不成立。时间复杂度 $O(N)$

C. String Transformation 1

题意:给两串字符串，每次可以经行一个操作，人选 a 字符串中一个字母 s 可以选 a 中一个或多个 s 将其变为比它字典序大的一个字母，问最少多少次操作可以实现把 a 字符串变成 b


题解：采取从低位到高位策略，这样可以尽量顾及到，连续变化削减次数的情况，即每次将所有 a 中的字符 i 都变成字符 j 直到没得变为止。因为变得顺序是从小字母到大字母，所以不会出现，变了之后 b 中字母反而小于 a 中字母的情况。可以采取符号维护 $f[i][j]$ 解决 f 表示 a 为 s_i , b 为 s_j 是否存在。

D GameGame

题意：给 n 个数，两个人进行博弈，一次去一个数，对自己现有的数去异或，问最后的输赢情况。

题解：

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforces_round_659_div2&rev=1596182742 

Last update: **2020/07/31 16:05**