20240416题解

A

直接贪，先做贵的面包，如果材料有剩再做另一种面包，统计总和就行。

B

暴力。

我们可以循环除边界以外的所有点，判断这个点以及这个点附近的点（见题目描述）在目标数组中是否为 1。如果它们都是 1，则将全零数组变为 1，并将当前的那个点存入答案数组。

操作过后，先看操作数组是否与目标数组相等，如果有一个点不相等，则无解，输出 -1。判完无解后先输出操作次数，再输出操作的点。

C

题目里要求铺设柱子和管子的总花费最小。

我们显然可以发现,当有交叉路口的时候,我们必须让这里的高度为2(废话)

但是在没有交叉路口的时候,我们可以让它的高度为2,也可以让它的高度为1.

所以dp方程就很显然了:我们设f[i][j]表示铺设到了前i的区域,当前铺设的高度为1/2时的最小花费。

那么转移方程就是

f[i+1][0]=min(f[i][0],f[i][1]+a)+a+b;f[i+1][1]=min(f[i][0]+a,f[i][1])+a+2\*b;(s[i]==′0′)

f[i+1][1]=f[i][1]+a+2\*b;f[i+1][0]=INF;(s[i]==′1′)

D

发现题意就是任意交换二元组，使得n个二元组不存在一维单调不减。

简单的转换：答案=总方案-单调不减的方案数

由于有两维，容斥，单调不减=第一维单调不减+第二维单调不减-两维都单调不减

单调不减可以分别按两维排序然后O(n)求出方案数。

那么单调不减如果有多种不同方案，就一定是两个相等的元素交换顺序。O(n)的统计就是统计相等的元素数量cnt[x]，并把答案乘上cnt[x]!即可。

E

首先，设答案为 X，每次挑选到的两个数为 A,B，返回的异或值为 C,D，则可以得到以下两个方程：X xor A=C，X xor B=D。从而得到：X xor A xor X xor B=C xor D。由于 X xor X=0，故 A xor B=C xor D。

因此，我们只要构造出两个序列，使其满足：对于任意 i,j，满足 Ai xor Bj唯一。就可以确定答案。

最简单的方法是：把第一个序列设为 [1,2,3,...,100]，第二个序列设为 [1×128,2×128,3×128,...,100×128]。这样子构造，第一个序列的前七位全为 0，第二个序列的后七位全为 0。二者不影响。

所以， 在询问到答案 C,D 后，可以直接得到 B=(C xor D) >> 7 << 7。输出 B xor D 即可。

F

显然的根号分治。

若询问的x>sqrt(n)，直接暴力跳就可以，这样跳的总个数不超过sqrt(n)。

否则，开一个数组，将所有(x,y)的结果打包预处理出来。由于x≤sqrt(n)，故数组规模是O(n)的。而对于修改，由于只需要修改不超过sqrt(n)个和，复杂度不超过O(q sqrt(n))。

综上，总复杂度O(q sqrt(n))。

G

AC自动机。

但是如果对s建立fail树，需要动态修改，不好搞。

所以我们对询问t建立fail树。那么这样就是在 ACAM 上跑 si，求出有多少个跑到的节点在 fail 树上以 t 的终止节点的子树中。这个可以对 fail 树进行一遍 dfs，用每个节点的 dfs 序和 size 维护。这样就是单点修改，区间查询，用树状数组即可。

可是 si的总长度可能会很大。不难发现每个 si形成了一个依赖关系，建出树，我们只需要再对这个 “操作树” 进行dfs，先计算贡献（位置 sonp,ci加上 1），再更新并下传跑到的位置 p=sonp,ci，最后撤销贡献即可。

时间复杂度O((n+m)log(sum(|t|))。