20240611题解

A

容易发现c[n]是在c[n-1]的基础上和乘以b[n]的项一起取最大，即c[n]=max{c[n-1],max{a[1],a[2],...,a[n]}\*b[n]}。但是直接计算还是O(n2)的，因此可以想到对a数组取一个前缀最大值将max{a[1],a[2],...,a[n]}预处理出来，这样的话就不需要比较，复杂度降至O(n)。

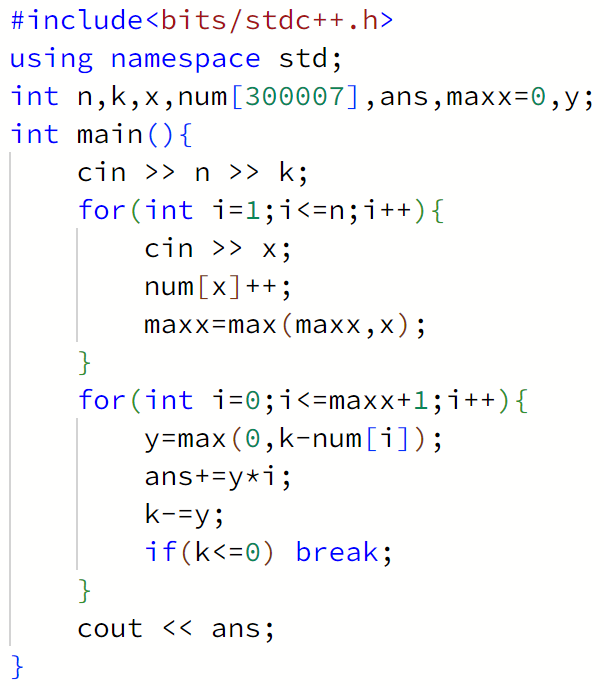
B

显然每个盒子都应该放置从0开始的连续整数才能使得答案最大。

那么我们开个桶统计每种数的出现次数，假如有t[0]个0，那么有max{0,k-t[0]}个盒子的mex就是0。

记k[i]为判断到i时还没确定最大值的盒子数量，假设有t[i]个i，那么有max{0,k[i-1]-t[i]}个盒子的mex为i，因此利用上面的方法循环一遍计数即可。注意循环到n+1。

以防有人看不懂，贴一份代码。



C

情况数极多，枚举是不可能的，考虑DP。

这里DP不是按每种填写情况的路径数量来做的，而是按每种路径可能的填写情况来做的。这也是所谓的“算两次”思想。

记dp[i][j][k]为从(1,1)走到(i,j)，经过了k个未指定符号的格子的总路径数。注意原本就有字符的格子在转移的时候只向对应方向转移。这么计算的话，最后总答案是，因为在路径上的格子只能填对应方向或者X，而不在路径上的格子可以任意填。

如果你会快速幂，恭喜你，直接秒了。O(HW+(H+W)log(H+W))能做。

如果你不会，在上面式子里把和k有关的项单独提出来，变成，因此整体每经过一个k将式子整体成2/3即可。这样可以避开快速幂。

D

构造题，想打表随便，反正N=8的表不好打。

翻译一下官解的构造。记s=2^(N-1)。

由于每次对战，不同队的人的对数会增加s^2，而因为我们需要让比赛场次最少，而最少的场次是(n+m)，因此在此条件下(n+m)s^2=m\*C(2s,2)。（等式左边是实际产生的对数，右边是理论上需要产生的对数）

将这条式子化简可以得到n:m=s-1:s，因此最少场数必是2^N-1。

构造方案可以是：对第i场比赛，如果popcount(i&j)为奇数，将这个人放进A队，否则放进B队。由于人数是2的次幂，可以证明这么放永远可以使A队和B队人数相等。而对于每一对(i,j)，假设二进制表示中存在一位使得i的这一位是1而j的这一位是0（不存在的话将i和j交换），记为第x位，那么对于k和(k^(1<<x-1))这两个数，它们按位与j的结果一样，但是按位与i的结果差一个1，那么这两场比赛里必有一场使得i和j同队，另一场i和j不同队。排除掉第0场（如果有此场，那么i和j同队），那么i和j不同队的次数就是2^(N-1)，同队的次数是2^(N-1)-1。

E

洛谷题解区：这真的是人能想出来的？

由于 Ant 只是按照一个固定的方式移动，看不到 Snuke 选了哪些物品。所以 Snuke 可以把自己的操作积累起来，等到 Ant 走到对应位置再操作，容易证明这样答案不变。

我们记 dp[l][r][k] 表示当前区间 [l,r] 已经被取完，Snuke 还有 k 步的余额，则每一步 Snuke 有以下两种选择：

1、自己选择一个数，可以选 l−1 或 r+1，得到该物品的权值，操作后k←k−1。

2、放弃自己的回合，让 Ant 选一个数。注意 Ant 只能选 l−1 和 r+1 中权值较大的那一个，操作后 k←k+1。

时间复杂度为 O(n^3)。

F

洛谷题解区：有点恶心。

先考虑没有 e 这样重复字符的另一个字符串，比如 abcdefg，设 pi为 ti所在的块，那么如果 t[i−1]<t[i]⇒p[i−1]≤p[i]，t[i−1]>t[i]⇒p[i−1]<p[i]，如果有 k 个小于限制，易得方案数为C(n+|t|−1−k,n−1−k)，该式子只含未知量 k。没有e的时候直接组合数做就行了。

再回到本题，由于存在 e 导致不清楚 e 具体位置所以难以确定 k，所以我们设 a[i]为使得 k=i 的方案数，设其生成函数为 。现在我们想直接求出 F。

这种多个低次多项式合并通常考虑分治，像线段树一样向上逐步合并，但是由于合并时与边界的状态（e 的具体位置）有关，所以需要额外记录下边界的状态，即 Fl,r,statel,stater(x)，合并时将中间状态相同的合并，即Fl,r,x,y(x)=Fl,mid,x,z(x)\*Fmid+1,r,z,y(x)。

复杂度 O(|t|log2|t|)，常数很大。

注意卡常，把你最快的 DFT 拿来用，有 e 的位置才记录多维。