2018 Multi-University Training Contest 9 Solutions

北京大学

视频题解 `https://www.bilibili.com/video/av29887345`

1001

恰有一个纯策略纳什均衡等价于只有 nm 这个数同时为所在行列的最大值。

考虑从大到小将所有数依次填入矩阵，令 $dp[i][j][k]$ 表示当前填入了最大的 i 个数，有 j 行 k 列已经放入了至少一个数。由于新填入的数不能同时同时为行列的最大值，因此转移有三种可能性：

1. 所在行列都已经有数了，这样可能的位置有 $jk-i$ 个，转移到 $dp[i+1][j][k]$。2. 只有所在行有数，这样可能的位置有 $j(m-k)$ 个，转移到 $dp[i+1][j][k+1]$。3. 只有所在列有数，这样可能的位置有 $(n-j)k$ 个，转移到 $dp[i+1][j+1][k]$。

时间复杂度 $O(n^2m^2)$。

1002

对于某一行，如果删第 i 个元素和第 j 个元素得到的结果相同，那么肯定有这一行 $[i,j]$ 内所有元素相同。

因此，矩阵的每一行根据连续出现的相同数被划分出了若干个等价类，每一个等价类是一个区间。例如行 0011100，就有三个等价类，分别为区间 $[1,2],[3,5]$ 和 $[6,7]$。

两条 Seam a,a' 的结果相同当且仅当对于每一行 i，$(i,a_i),(i,a'_i)$ 都在同一个等价类内。

这题有很多时间复杂度为 $O(nm)$ 的动态规划做法，这儿就直接讲原来标算的做法了。

对于一个 Seam 的等价类 S，S 中的所有 Seam 最后一行的值一定构成了一个区间 $[l,r]$。令 $f[i][l,r]$ 表示只考虑前 i 行，最后一行区间为 $[l,r]$ 的等价类个数。

转移即枚举第 $i+1$ 行的一个等价类 $[l_i,r_i]$，$f[i][l,r]$ 可以转移到 $f[i+1][[l-K,r+K]\cap[l_i,r_i]]$。这样可以得到一个 $O(nm^3)$ 的 DP。

可以发下虽然 $[l,r]$ 可能可以转移到很多个等价类，但是其中绝大部分都是覆盖整个等价类的，只有最左端和最右端可能覆盖非整个的等价类，因此转移可以用前缀和优化到 $O(1)$。

最后可以发现一个结论，上述 DP 的有效状态是 $O(nm)$ 的，即有值的状态 i,l,r 只有 $O(nm)$ 个。这个结论可以由下述事实直接得到：对于每一行 i，任意两个有效状态 $[l_1,r_1],[l_2,r_2]$ 不可能有 $l_1<l_2<r_2<r_1$ 的位置关系。

上面这个结论直接归纳即可。因此只需要再有效状态之间更新就可以了，时间复杂度 $O(nm)$。

1003

定义 a,b 之间的差异集合 $S_{a,b}$ 为所有满足在 a,b 中指数不同的质数 p 形成的集合，那么 $d_{a,b}$ 的取值为：

1. $\sum_{p\in S_{a,b}} p$，当 $S_{a,b}$ 非空时。2. 1，当 $S_{a,b}=\emptyset$ 时。

现在对着两种情况分别计算，对于质数 p，令 A_p 表示 $[1,n]$ 中 p 指标为奇数的数的个数，那么 A_p 可以在 $O(\log n)$ 的时间内计算得到：

$$\sum_{a,b,|S_{a,b}|>0} d_{a,b} = \sum_{p=1}^{n} p \times A_p \times (n - A_p)$$

以 $\sqrt{n}$ 为分界点，小于 $\sqrt{n}$ 的质数可以暴力枚举计算，时间复杂度 $O(\sqrt{n})$。

对于更大质数，其指标不可能超过 1，因此必有 $A_p = \lfloor \frac{n}{p} \rfloor$。这时 $A_p \times (n - A_p)$ 只有 $O(\sqrt{n})$ 种不同的取值，每一种取值对应了一个 p 的区间 $[l_i, r_i]$，对每一个区间计算内部所有质数的和即可。这可以用洲阁筛在 $O(n^{\frac{3}{4}}/\log n)$ 的时间复杂度内解决。

对于第二种情况，考虑两个 $S_{a,b}=\emptyset$ 的数 a,b，它们必定可以被表示成 $a = xc^2, b = xd^2$ 的形式，其中 x 满足 x 中无平方因子，即 $|\mu(x)| = \mu(x)^2 = 1$。

因此对这一部分求和为：

$$\sum_{a,b,|S_{a,b}|=0} d_{a,b} = \sum_{x\in[1,n]} \mu(x)^2 (\lfloor \sqrt{\lfloor \frac{n}{x} \rfloor} \rfloor)^2$$

和上一步一样，后面部分的取值只有 $O(\sqrt{n})$ 种，因此只要计算 $\mu(x)^2$ 的前缀和就行了。这是一个经典的问题：

$$\begin{aligned}\sum_{i=1}^{n} \mu(i)^2 &= \sum_{i=1}^{n} \sum_{d^2|i} \mu(d) \\ &= \sum_{d=1}^{\sqrt{n}} \mu(d) \lfloor \frac{n}{d^2} \rfloor.\end{aligned}$$

暴力枚举 d 计算就行了，这样第二部分也可以顺利解决。总的时间复杂度也为 $O(n^{\frac{3}{4}}/\log n)$。

1004

结论：无论六花采用怎样的决策，其期望收益始终相同。

在这个结论下，我们考虑六花也随机出牌的期望收益：六花出剪刀的时候，有 $\frac{b}{n}$ 的概率碰到石头，$\frac{c}{n}$ 的概率碰到布，因此剪刀的期望收益和为 $\frac{a'c-a'b}{n}$。

其他类型的卡片同理，可以得到六花总的期望收益为 $\frac{a'c-a'b+b'a-b'c+c'b-c'a}{n}$。

证明可以用简单的归纳法，令 $F(n,a,b,c,d)$ 表示六花有 a 张剪刀，b 张石头，$n-a-b$ 张布，勇太有 c 张剪刀，d 张石头，$n-c-d$ 张布时六花的最优收益。归纳证明对于所有 n,a,b,c,d，六花不管怎么出，期望收益都和随机出一样。

当 $n=1$ 时显然成立。

当 $n>1$ 时，我们不妨假设六花出的是剪刀，根据归纳假设，此时的收益为：

$$\frac{c}{n}f(n-1,a-1,b,c-1,d)+\frac{d}{n}(f(n-1,a-1,b,c,d-1)-1)+\frac{n-c-d}{n}(f(n-1,a-1,b,c,d)+1)$$

完全展开这个式子，可以得到与随机出的期望式子相同，因此归纳成立。

时间复杂度 $O(1)$。

1005

所有能够躲雨的点一定形成了平面上若干个封闭区域。询问相当于询问一个点到这些区域的最近距离。

点 p 能够躲雨当且仅当满足如下两个条件：1. p 在多边形内。2. p 到多边形边界的最近距离大于等于 R。

因此可以在 $O(n)$ 的时间内判断一个点是否合法。

考虑封闭区域的边界来自于哪里，只有两种可能性：1. 以多边形顶点 x_i 的圆心半径为 R 的圆的圆周。2. 多边形的边 (x_i, x_{i+1}) 向内平移 R 得到的直线。

这样边界有 $O(n)$ 个来源，边界的顶点只可能来源于这 $O(n)$ 个元素两两之间的交点，共 $O(n^2)$ 个。暴力判断每一个交点是否合法，这样能得到所有的顶点集合 S（确切说是顶点的超集），时间复杂度 $O(n^2)$。

考虑一个点到这些封闭区域的最近路线有哪些可能性：1. 这个点到边界顶点的连线。求 S 内所有点到询问点的最近距离即可 2. 这个点到边界一条线段的垂线。依次求询问点到 $O(n)$ 条直线的垂足，如果垂足合法，那么用最短距离更新答案 3. 这个点到边界圆弧的最近点。依次求询问点到 $O(n)$ 个圆的最近点，如果最近点合法，那么用最近点更新答案。4. 点本身就在封闭区域内。判断询问点是否合法，如果合法答案为 0。

这样我们可以在 $O(n^2)$ 的时间内处理一组询问，总时间复杂度 $O(n^3+mn^2)$。

值得注意的是这个问题精度非常敏感。边界上的每一个点到多边形边界的最近距离都是 R。如果 eps 设的过小，那么很容易把边界点判成不合法，答案会偏离非常多。

正确的处理方法是设置一个非常大的 eps 比如 10^{-3}。因此当 eps 过大的时候，如果把原来不合法的点 p 当成合法，那么 p 至少也是半径 $R+\text{eps}$ 的合法点。题目保证了 R 加减 0.1 答案不变，因此这样不会改变答案。

1006

对于任意图，求生成树个数可以用矩阵树定理转化成求基尔霍夫矩阵的行列式，时间复杂度 $O(n^3)$。

在这题中虽然图中的点很多，但是团的个数和额外的边数只有 200。即最多只有 400 个点（关键点）是和外界有联系的，剩下的绝大部分点都只和团内部的点有连边，是完全等价的。如果可以把每一个团中的非关键点给缩小到一个，那么图中就至多只有 $2m+n\le 600$ 个点，可以跑高斯消元了。

具体的思路是把图的基尔霍夫矩阵写出来，可以发现几乎是一个分块对角矩阵 (只有关键点的行列和其他块之间有边)。可以对每一个块的部分矩阵做初等行列变换然后把非关键点所在的行列都展开。具体可以看视频题解部分的详细讲解 (在文字题解里画矩阵太麻烦了 qwq)。

时间复杂度 $O(n^3)$.

1007

树上关于路径的问题是有一个经典的关于**线头**的 DP 的。考虑一个方案在以 i 为根的子树下的截面：有一些路径完整的在子树内，还有一些路径只有部分在 i 子树内，它们在截面中必然有一个端点为点 i，把这种部分的路径叫做线头。

因此令 $dp[i][j][k]$ 表示以 i 为根的子树，有 j 条完整的路径，k 个线头。在把孩子 c 接到父亲 k 上时，考虑合并 $dp[i][j][k], dp[c][a][b]$。由于 i 的线头和 c 的线头之间可以合并，枚举合并 m 个线头，则应当转移到 $dp[i][j+a+m][k+b-2m]$，同时由于余下的 $b-m$ 个线头都变长了，因此还要支付 $(b-m)C$ 的代价。

这样可以得到一个复杂度非常高的 DP，优化分为如下两步。

首先可以证明树上每一条边最多被经过两次：如果被经过了超过两次，那么可以选择两条横跨的路径，让左边两个端点匹配，右边两个端点匹配，从而在不改变节点经过状态的情况下，缩减路径的总长度。

因此第三维状态 k，线头数只需要记 $0,1,2$ 三种值，这样状态数就减少到了 $O(n^2)$。

接着，可以发现一个大小为 s 的子树内，最优解至多只需要 s 条完整的路径，这样就可以保证每一个点都被单独的经过一次，收益达到理论最大值。因此状态 j 只需要记到 $size_i$。这个优化的好处在于可以将一次合并的代价减少到两个子树大小的乘积。这是一个经典的 DP 形式，时间复杂度为 $O(n^2)$。

这样我们可以在 $O(n^2)$ 的时间内解决这个题。具体的证明和图示可以见视频题解。

1008

这个题就是个大分类讨论题。考虑怎么将 $L(L(T))$ 上的点/边/路径对应到 T 上。

$L(L(T))$ 的每一个点对应了 T 上的一条三个点的链，每一条边相当于把两端的一个点删了再换一个点上去。

因为在树上两个点之间的最短路是唯一的。简单讨论一下可以发现线图上点 (a,b,c) 和 (d,e,f) 之间的最短路可以表示成三部分的和。令点集 $\{a,b,c\},\{d,e,f\}$ 之间唯一的最短路连接的是 x,y 两个点，那么最短路的三部分为：1. 关于边 $(a,b),(b,c)$ 权值以及 x 位置的函数。2. 关于边 $(d,e),(e,f)$ 权值以及 y 位置的函数。3. 路径 (x,y) 长度乘以 4。

针对性地用 treedp 求解即可。同时还要讨论 $(a,b,c),(d,e,f)$ 在某种程度上重叠的情况。

时间复杂度 $O(n)$。因为更详细的讨论需要画图，用文字题解不方便，可以见视频题解。

1009

先考虑一个简单的问题：对一个序列 A 做 K 次冒泡排序，如何计算结果序列?

对于 A_i，在一次冒泡排序后，它可能会向后移动任意长的距离，但是最多只会向前移动一位。因此结果数组 B 的第一位一定是 A 中 $[1,K]$ 中的最小值，第二位一定是 $[1,K+1]$ 中除去 B_1 的最小值。以此类推，第 i 位是 $[1,K+i]$ 中除去 B_1 至 B_{i-1} 的最小值。

用一个堆就可以维护了，时间复杂度 $O(n\log n)$。

考虑对序列分块。对于一个单独的块 B，考虑如果以此冒泡排序完整的覆盖了 B，会对 B 产生什么影响。具体来说可以分为 3 步：1. 会询问 B 的开头元素，看能否与之前块带过来的数 x 进行交换。如果能交换，会修改开头元素。2. 会对 B 做一次整体的冒泡排序。3. 会询问 B 在冒泡排序结束后的末尾元素，即在冒泡排序前 B 中的数的最大值。

因此块 B 内要进行的操作是：删除开头元素，在开头插入元素（之后必定伴随一次冒泡排序），修改末尾元素，做整体冒泡排序。要进行的询问是：询问最大值，询问开头元素，询问整体的和。

对于整体和，由于必须要维护最大值和开头元素，因此每一个插入删除操作时涉及到的元素是已知的，直接加减就行了。

对于最大值来说，操作相当于对于一个集合插入/删除元素，询问最大值。只需要对每一个元素记录当前有没有被删除，用堆维护就行了。

对于询问开头元素，只需要维护所有“可能是”开头元素的所有数的最小值就可以了。最开始，可能作为开头元素的元素只有块的初始左端点。每一次删除一个元素或者进行了一次冒泡排序，候选区间的右端点应当加一，即纳入一个新的元素进来。

由于插入元素一定伴随着冒泡排序，因此对插入元素 + 冒泡排序的整体操作来说，候选区间的右端点应当保持不变，但是新加入的元素应当加入可能集合中。

注意，修改末尾元素带来的新的数是有可能被移动到开头的。在被整体做 k 次冒泡排序后插入的元素可以视为原来处在区间右端点 $+k$ 下标的元素，在足够多次整体冒泡排序后仍然会被纳入候选区间。

因此对于开头元素来说，要做的仍然是对集合插入删除元素询问最小值。还是可以通过堆来解决。

因此对于块的整体操作可以在 $O(\log n)$ 的时间内解决，与块的长度并无线性关系。

接着如果询问和冒泡排序部分地设计到了一个块，就要把这个块重构，重构一个块可以看成重复地使用询问左端点，删除左端点操作，可以在 $O(L\log n)$ 的时间内解决，其中 L 为块长度。

综上，如果对序列分块，每一块大小为 S，时间复杂度为 $O(mS\log n + m\frac{n}{S}\log n)$，取 $S=\sqrt{n}$，时间复杂度为 $O(m\sqrt{n}\log n)$。

标程采用了对序列分块的写法，这样常数更小也更方便。即对询问序列分块，每一块大小为 S，块内的询问只会有 $2S$ 个端点，把序列分成了 $2S+1$ 块。对每一块用上面提到的数据结构进行维护，这样操作只会涉及到完整的块。在每一个询问块结束后，需要把整个序列重构。

因此，如果对询问分块，时间复杂度为 $O(mS\log n + \frac{m}{S}n\log n)$，仍然取 $S=\sqrt{n}$ 得到时间复杂度为 $O(m\sqrt{n}\log n)$。但常数和细节都要更少些。

1010

为了题解方便，这儿假设两个序列长度都是 3。在长度不足 3 的时候，你可以用 $+\infty$ 把序列补足 3 并假设 $f_{+\infty}(n)$ 在 n 趋于无穷的时候为 1。

把 $f_x(n)$ 简写作 f_x，我们要判断 $f_{A_1}^{f_{A_2}^{f_{A_3}}}$ 和 $f_{B_1}^{f_{B_2}^{f_{B_3}}}$。取两次 log，我们需要比 $f_{A_1+2}+f_{A_2+1}f_{A_3}$ 和 $f_{B_1+2}+f_{B_2+1}f_{B_3}$ 哪个大。

根据尝试，我们需要将 f_{A_1+2} 和 $f_{A_2+1}f_{A_3}$ 中的较大值和 f_{B_1+2} 和 $f_{B_2+1}f_{B_3}$。但是需要注意的是，由于我们取了两次 log，因此在较大项比不出大小的时候，应当再比较余项，即比较较小值之间的大小。

这么做的原因举例来说，考虑 e^{n^2+n} 和 e^{n^2}，虽然在取 log 之后，n^2+n 和 n^2 同阶，但是 $\lim_{n\to+\infty}\frac{e^{n^2+n}}{e^{n^2}} = \lim_{n\to+\infty} e^n = +\infty$。

为了方便，我们可以利用 $f(+\infty)$ 的假设统一形式，即把 f_{A_1+2} 看做 $f_{A_1+2}f(+\infty)$。这样就只需要考虑比较 $f(a)f(b)$ 和 $f(c)f(d)$ 即可。显而易见，他们的大小关系等价于先比 $\min(a,b), \min(c,d)$ 的大小，如果相同，再比较 $\max(a,b), \max(c,d)$ 的大小。

这样可以在 $O(1)$ 的时间内回答。(感觉大家不能沉迷 ACM 忘了专业课啊？)

1011

签到题，怎么做都行。可以进行分类讨论，也可以使用简单的 DP。

标程使用的 DP 状态为 $f[a][b]$，a 表示球的数量对 1 取 min，b 表示拍的数量对 2 取 min，直接对四类人更新一下一下就可以了。

要注意的是运算涉及到 2^a，你可以预处理出 1 到 10^7 的 2 的幂，也可以用快速幂算。

单组时间复杂度 $O(1)$ 或 $O(\log a)$。