2021 ICPC Asia Jinan 题解

北京大学命题组

2021年11月14日

Space Station

题意

- 现在有一棵在三维空间中的树，树上每条边一定是某个坐标轴(x/y/z)正/反方向的向量。
- 现在有两类操作。
- 旋转操作：u v axis degree。 axis=x/y/z,degree=90/180/270。 表示将边(u,v)以从u出发向axis的正方向向量为轴，旋转degree度。
- 询问操作：u v。你需要回答u,v之间的欧式距离。
- n,Q<=1e5

Space Station

题解

- 建立边权为(x,y,z)相对坐标关系的有根树，并用线段树维护区间和，操作1分为两类，一类是“父亲固定，旋转儿子”，对应儿子所在子树的区间乘，乘的是一个表示旋转方向的3x3矩阵，另一类是“儿子固定，父亲旋转”，对应儿子所在子树的区间乘 + 儿子单点乘 + 整棵树的区间乘；操作2就是树上两点间的距离(sumx,sumy,sumz)。树链剖分，时间复杂度 O(Q*logN*logN)。也可以按dfs序建线段树做到O(Q*logN)。

Monitor Area

题意

- 给一个建筑(简单多边形)和其内部的一些摄像头(点)，求监控面积。
- 一个建筑内的点能被监控当且仅当存在一个摄像头到该点的线段全部在建筑内。
- $n, m<=50$

Monitor Area

题解

- 首先，对于每个摄像头，按极角做一遍扫描线，对于每个角度求出距离最近的边，这样它能监控的区域就可以用O(n)个三角形的并来描述。
- 之后要求O(nm)个三角形的面积的并：
- 法一：求出两两边的交点后，按x坐标做一遍扫描线即可。直接做的话时间复杂度可能是O((nm)^3 log(nm)) (但也可以通过)，可以用平衡树优化到O((nm)^2 log(nm))；
- 法二：扣出边界后格林公式。这里有个细节问题是边界存在重边的情况时只能计算一次，只需规定个顺序每次只让最前面的计算即可。时间复杂度O((nm)^2 log(nm))。

Optimal Strategy

题意

- 有 n 件物品，第 i 件的价值为 a[i]。A 和 B 轮流取物品，A 先手。每个玩家都要最大化自己取到的物品的价值和，求有多少种可能的游戏过程。
- 1 <= n <= 1 000 000
- 1 <= a[i] <= n

Optimal Strategy

- 考虑最大值如果是偶数个，那么会每次被两个两个的取；如果是奇数个，那么会被先手立刻取走一个，变成偶数的情况。
- 故容易得到答案为（c[i] 是价值为 i 的物品个数）

$$\prod_{i=1}^{n} c_i! \binom{\sum_{j=1}^{i-1} c_j + \lfloor c_i/2 \rfloor}{\lfloor c_i/2 \rfloor}$$

Insidemen

题意

- 圆环上的n个点之间有m条连边，若两条边(a,b)(c,d)在圆内相交则产生(a+b)*(c+d)的权值。挑选两点删除与其相连的所有边令剩余权值最大化。
- 1 <= n <= 10^3 1<=m<=10^5

Insidemen

题解

- 取定其中一个点后，对其他边可以用一维前缀和查询。
- 从而可以求出与某一个点相关的所有权值和，进而求出删点之前的权值和。
- 由于要删除两个，还要减出重复的部分。
- 时间复杂度O(NM)。

Arithmetic Sequence

题意

给你长度为n的整数数列a，请你使用整数等比数列拟合a，使得每个位置差的绝对值求和最小。

$n<=2e5$，$|a_i|<=10^{13}$

Arithmetic Sequence

题解

对于给定的公差 d，设其最小的代价是 f(d)。

f(d)的求解是个经典问题，只用nth_element求出中位数即可。

我们发现函数 f(d)是凸的。

因此，我们可以使用二分斜率来计算最优的 d。时间复杂度为 O(nlogW)，其中 W 是值域。

Arithmetic Sequence

题解

证明：我们可以通过 d-1 和 d+1 的答案来构造 d 的答案，构造的方法是取 d-1 和 d+1 的结果序列的平均值。我们可以看到，对于每个位置，2 * d的代价 <= d-1 的代价 + d+1的代价。对每个位置求和，可以得到 2*f(d) <= f(d-1) + f(d+1)。易证函数 f 是凸函数。

下面唯一的问题是我们用这种方法有可能构造出一个非整数解。显然如果解非整数，那么其中每个数必然形如 x + 0.5，其中x是整数。所以我们将每一个数加 0.5 或者减 0.5。很容易证明一定有一种方法可以满足上面的不等式。

Neural Network Counting

题意

神经网络为无向图，图中节点为算子，边表示数据对应的传递，数据为二维的矩阵。假设有三个算子：输入、乘法和卷积（作了相应的简化），它们定义了数据的计算。我们将神经网络的输入格式定义为神经网络的输入数据的形状（即输入的矩阵的形状，或者说每一维的长度）。

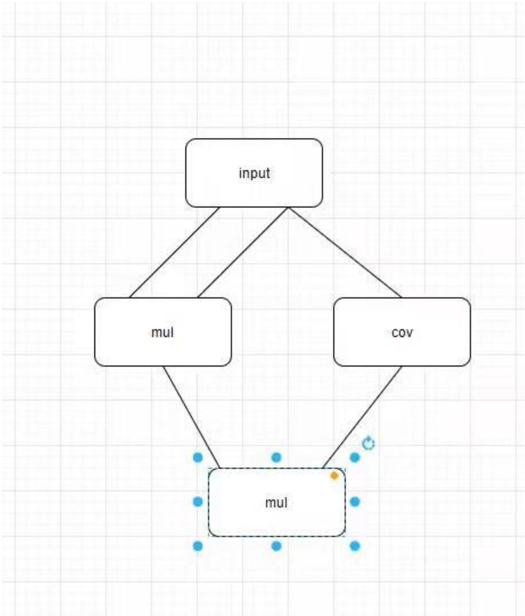
给定N和D。统计算子个数为N且输入数据每一维不超过2^D-1的神经网络有多少种（算子组合相同但输入格式不同算作不同的方案）。

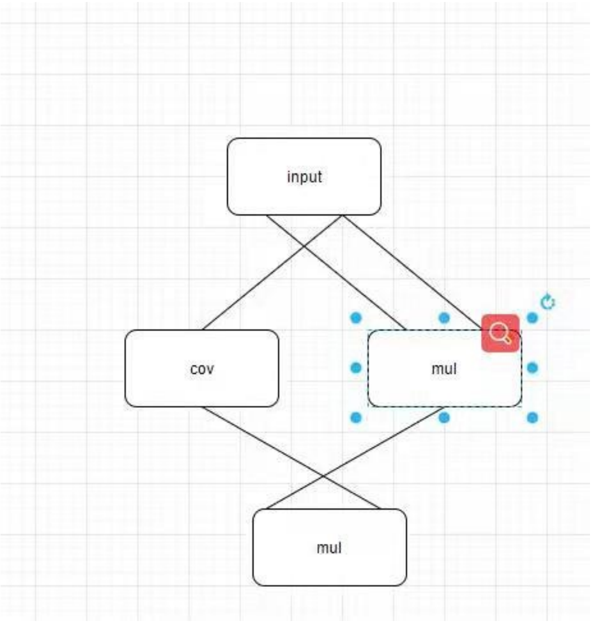
Neural Network Counting

题解

首先由于N比较小，可以暴力枚举出所有的神经网络结构。

这里需要注意同构的问题，如下图所示。





Neural Network Counting

题解

接下来可以将各自节点中的输出矩阵的形状看成两个变量，一共2n个变量。

对于卷积和乘积两种运算，我们可以将其抽象为变量间的约束关系，我们需要统计满足所有约束的变量赋值方案数。

约束分两种：1、相等关系；2、x = floor(y/2)

我们首先抽取所有相等关系，利用并查集合并变量。

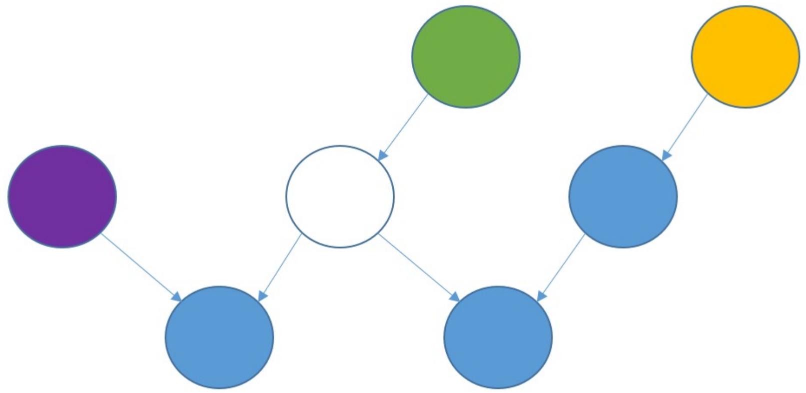
对于剩下的约束，我们将变量看成点，变量的关系看成有向边，那么就组成了一个分层图。我们需要做的就是对图中入度为0的点进行赋值。

我们先找到一个深度最深的点进行赋值（“/2”的次数最多的），对于剩下的需要赋值的点，我们需要计算其“多少个/2”之后那个点已经被赋值，最后方案数*（2^这个深度）。

Neural Network Counting

题解

举例来说，对于如下的变量关系图（边(x,y)表示y=floor(x/2)）：



我们先标记黄色节点，这样所有的蓝色节点的值都被黄色节点所确定了。对于绿色节点，我们计算出其离“确定的点”（蓝色点）的"/2"的次数为2，那么绿色节点会贡献2^2这样一个系数（黄色节点确定后，绿色节点可以取4种）。类似地，紫色节点会贡献2^1这样一个系数。

Happy Alice

题意

- 给一个序列，每个元素有价值v和颜色c。m次询问，每次询问一个区间[l,r]内，满足c[i]=c[j]，v[i]<v[j]且不存在l<=k<=r且满足c[i]!=c[k],v[i]<v[k]<v[j]的最大的v[j]-v[i]
- n<=10000,c={0,1},v<=10^9,m<=1000000

Happy Alice

题解

- 对于每个询问建点，使用kd-tree维护。
- 将每个元素按v从小到大排序，视作操作，在kd-tree上处理所有包含它的询问（是一个矩形）。
- kd-tree上维护lazy标记，包括子树操作序列前缀颜色，后缀颜色，同色前缀最后一个元素v，同色后缀第一个元素v，序列中最长同色操作段。
- 对于节点维护最长同色操作段，后缀颜色，同色后缀第一个元素v。
- 处理完所有标记后down下所有lazy标记得到答案。
- 时间复杂度O(nsqrt(m)+mlogm)

Game Coin

题意

Bob 第 i 天需要 $a[i]$ 个游戏金币来满足自己的游戏购物欲望。获取金币有两种方法，第一种方法是每个金币可以直接花费 t 元购买，第二种方法是购买游戏商城内部的金币卡来每天领取临时金币，总共有m 种金币卡，第j 种金币卡有一个购买价格$c[j]$，使用期限$d[j]$ 和每天的金币领取数量 $w[j]$ 。临时金币只能当天使用，第二天就失效。

倘若第i 天花费$c[j]$ 元购买了第j 种金币卡，那么从第i 天到第$i+d[j]-1$ 天每天都可以用该金币卡来领取 $w[j]$ 个临时金币。**每种金币卡可以无限数量购买，但是每次购买新的卡会把手上已有的卡给覆盖掉，保证手上最多只有一张卡。**（例如，第x 天买了一张金币卡i，可以用来领取$w[i]$ 个临时金币，然后购买一张金币卡j（这时会覆盖掉金币卡i），可以继续用新卡来领取$w[j]$ 个金币，通过这种方式第x 天可以获得$w[i]+w[j]$ 个金币，并且第 x 天结束时，手上有一张金币卡j）

计算满足Bob n 天的游戏购物欲望最少需要花多少钱。

$n \le 100000, m \le 400, \sum a[i] \le 500000$

$c[j], d[j], w[j] \in [0, 10^9]$

Game Coin

题解

- 首先可以用背包dp出一天通过买卡和单独购买的方式获取i个金币的最小代价 $f[i]$ (这里要注意求出来的f可能不是单调递增的，需要求一个后缀的极值)。
- 考虑 $d[i,j]$表示第i天最后剩下了满血j卡并且1~i天的金币需求全部满足所需要的最小代价($j=0$ 表示没剩下卡)。
- 转移的话可以考虑枚举d[k,p]（k<i,1<=p<=n,d[p]>i-k）

$$d[i,j] = min_{k<i,d[p]>i-k}(d[k,p] + c[j] + f[a[i] - w[p] - w[j]] + offset(k+1,i-1,p))$$

$$offset(i,j,p) = w\sum_{x=i}^{j} max(0,a[x] - w[p])$$

- 这样实现的时间复杂度为 $O(m^2n^2)$

Game Coin

题解

- 可以对每种金币卡 p 分别用单调队列优化掉 k 的枚举。
- 现在复杂度变成了 $O(m * n^2)$

Game Coin

题解

继续考虑$\sum a[i] < 500000$ 这个条件。对于第i 天，可以预处理出通过之前买的金币卡到今天不花钱能兑换到的金币u的最小代价$S[i,u]$，这样计算的时候可以不用再枚举p，而是改为枚举金币数量 u

$$d[i,j] = min_{u<=a[i]}(S[i,u] + c[j] + f[a[i] - u - w[j]])$$

- 这样时间复杂度就是$O(m * n + \sum a[i] * n)$

Permutation Pair

题意

PTY got a positive integer N and an array A with indices from 1 to N. He needs to count the total number of **ordered permutation pair** (P, Q) (i.e., an ordered pair (P, Q), where both P and Q are permutations of $\{1, 2, \cdots, N\}$) that satisfies the following conditions:

1. M is an $N \times N$ matrix where $M_{i,j} = \begin{cases} 1 & \exists k \in \{1, 2, \cdots, N\}, i = P_k \land j = Q_k \\ 0 & \text{otherwise} \end{cases}$.

2. For every $M_{i,j} = 1$, there is **NO** $(A_j + 1) \times (A_j + 1)$ submatrix of M (i.e., a subregion of M in consecutive $(A_j + 1)$ rows and $(A_j + 1)$ columns) that **covers** $M_{i,j}$ and is an identity matrix or horizontal mirror of the identity matrix.

Because of the large answer, you just need to tell PTY the answer mod 998244353. Can you help him?

Input

Two lines. The first line contains an integer N $(1 \le N \le 5000)$. The second line contains N positive numbers A_j $(1 \le A_j \le N)$ for j in $1, 2, \ldots, N$.

Permutation Pair

题解

本题等价于求排列P的个数，满足P中i所在连续公差为1的等差数列长度<=a[i]，乘n!即为最终答案。

考虑先将1~n做合法分段，再考虑左右翻转，打乱顺序。但我们要求是相邻段之间不能连起来。因此进行一个容斥操作：不妨计算至少有k个间隔连起来的方案数，乘以(-1)^k的容斥系数。通过二项式定理不难验证最终没有连续间隔的方案被计为1，其他非法方案计为0。

现在问题即变为如何求解强制k个间隔相连，其余间隔不作限制的方案数。通过一个递推就可以解决。

设f[i][j][0/1]表示当前已经将1~i分段，最终在方案中可以自由移动摆放的有j段，当前i所在的段是否是该自由段（一个自由段由若干个小段组成）的最后一段。转移即枚举当前i的小段，考虑要不要与上一段连续摆放（即合并到上一个自由段）每当添加一处连续/闭合一段，就结算他的系数：只有当自由段的长度>1，才会有左右翻转的系数2。最终乘上自由摆放的段数的阶乘，求和即可。

此递推容易通过前缀和优化到O(n^2).

Determinant

题意

- 给定一个矩阵， 并且给出它的精确行列式的绝对值
- 判断行列式的正负
- n<=100

Determinant

题解

- 随机一个大模数 P，令其不是给定行列式的因数。
- 计算行列式模 P 的值，与给定的数比较即可。
- 不难证明在 P 是奇数时行列式为正与为负的情况算出来的行列式总是不等。

Search For Mafuyu

题意

- 给定一棵 n 个点的树，A 在 1 号点，B 的位置在 2-n 中均匀随机，A 不知道 B 的位置。现在 A 要去找 B，每秒可以走到一个相邻点，求在最优策略下的期望时间。
- 1 <= t <= 1 000
- 1 <= n <= 100

Search For Mafuyu

题解

- 先假设走的路径是一个欧拉遍历。容易发现交换一个点的两棵子树答案不变。
- 有了上述结论之后，容易证明最优解确实是一个欧拉遍历，也就是不会半途返回。
- 按任意顺序 DFS 即可。时间复杂度 $O(n)$。

Strange Series

题意

- 给定 n 和次数不超过 n 的整系数多项式 f(x)，计算 $\sum_{i=0}^{\infty}\frac{f(i)}{i!}$ 。可以证明答案是 e 的整数倍，只需输出这个倍数对 P 取模的结果即可。
- 数据范围：1 <= n <= 100000, P = 998244353

Strange Series

题解

- 我们设 a_n 表示 f(x) = x^n 的答案，考虑计算这个数列的指数生成函数：

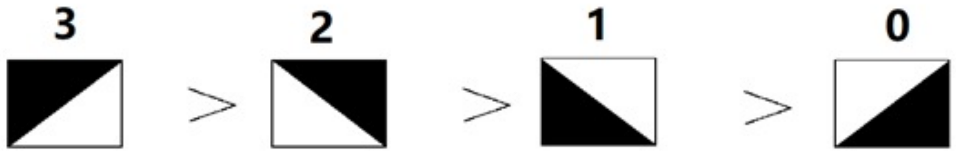
$$g(x) = \sum_{n\ge0}\frac{x^n}{n!}\sum_{i\ge0}\frac{i^n}{i!} = \sum_{i\ge0}\frac{1}{i!}\sum_{n\ge0}\frac{(ix)^n}{n!} = \sum_{i\ge0}\frac{e^{ix}}{i!} = e^{e^x} = e^{e^x-1}e$$

- 于是使用一次多项式 exp 即可算出答案，时间复杂度 O(n log n)。值得一提，a_n 恰好是 Bell 数，这也就证明了答案是 e 的整数倍。

Coloring Rectangles

题意

- 在直角坐标系中给出 N 个空白矩形
- 对每个矩形需要把它沿着某一条对角线分割成两半，并把其中一半涂黑
- 求字典序最大的涂黑方案，使所有黑色三角形不重合（除边、点相交外）
  - 按如下编码生成方案编码后比较字典序:



- $N \leq 200$

Coloring Rectangles

题解

- 先把矩形按照两条对角线划分成四个小三角形
- 考虑任何一种涂黑之后的部分，其必由两个小三角组成
- 把面对面的两个小三角分为一组，共分成两组
- 发现必须在每组三角形中恰好选一个涂黑
- 于是问题转化为 2n 个变量的 2-SAT，若小三角形相交 则不能同时选
- 划分的字典序恰好可以看做 2n 个变量并有相应的字典序
- 建立 2-SAT 后求字典序最小的解即可
- 时间复杂度O(n^3)