2019 年西安电子科技大学程序设计竞赛网络预选赛非官方题解

xry111

April 20, 2019

写在前面

一些废话

- 普及和宣传不是比赛水的理由，题难不是违纪的理由。参赛人数很多且纪律涣散的比赛我们学校已经有一个了，我们不想做第二个。
- 奇怪的技巧不是不可以用，但请首先把基础打好，不要没学会走路就想跑甚至倒立着跑。

难度

- 难度 1 的题目是区域赛中几乎所有队伍都能做出来的题目，如果你没有通过所有难度小于等于 1 的题目，说明你不适合打这个比赛，可能应该重修 C 语言。
- 如果你能通过所有难度小于等于 2 的题目，那么恭喜你，你应该能在区域赛中获得铜牌。
- 如果你能通过所有难度小于等于 3 的题目，那么恭喜你，你应该能在区域赛中获得银牌。
- 如果你能通过难度 4 的题目，那么你有可能在区域赛中获得金牌。然而这场比赛并没有那么难的题，毕竟我们不能指望通过校赛选拔金牌选手。

Problem A　上帝视角

难度　1

关键字　模拟

题解　把所有筹码加起来就是总的筹码数，如果它不能被人数 n 整除，说明局面是不合法的（因为这样一开始就不可能所有人筹码数相同）。否则，把总的筹码数除以 n 就得到初始筹码数，然后对于每个询问直接判断即可。

时间复杂度 $O(n+m)$。

Problem B　Shocking! Two Acmer Doing This In The Lab!

难度 1.5

关键字 分类讨论

题解 考虑最大/最小化 wang9897 的分数的情况（对于 qkoqhh 把他俩的分数倒过来即可）。首先考虑如何最大化 wang9897 的分数，容易看出，此时他要么一局都没有输，要么只输了一局且 qkoqhh 在这一局中 AC 了全部 m 道题（这样其他局就都是 wang9897 赢或者平局）。对于第二种情况，把局数 k 减掉 1，m 变为 0，就转化成了第一种情况。第一种情况只有 $n \geq m$ 时才成立，此时 wang9897 赢的局数 $w = \min\{n-m, k\}$，平局的局数就是 $k-w$，所以得分是 $3w + (k-w) = 2w + k$。

再考虑如何最小化 wang9897 的分数。同理，这里也只有两种情况，要么他一局都没有赢，要么他只赢了一局并且在这一局中 AC 了全部 n 道题目。第二种情况同样可以转化成第一种情况，第一种情况只有 $n \leq m$ 时成立，此时 wang9897 输的局数 $l = \min\{m-n, k\}$，平局局数为 $k-l$，所以 wang9897 的得分就是 $k-l$。

时间复杂度 $O(1)$。

Problem C　XY 之说走就走的旅行

难度 1

关键字 广度优先遍历

题解 从 X 和 Y 开始分别进行广度优先遍历，得到每个点距它们的距离，然后枚举所有 P 即可得到答案。

时间复杂度 $O(nm)$。

Problem D　武举考试安排表

难度 2

关键字 递推，分治

题解　对于 $n=1$ 的情况（即有 2 名选手），答案很简单，就是矩阵 $\mathbf{A}_1=[2,1]$。我们通过找规律可以看出，能够从 $n=k$ 的情况推出 $n=k+1$ 的情况：

$$\mathbf{A}_{k+1}=\begin{bmatrix}\mathbf{A}_k & \mathbf{A}_k+2^k\times\mathbf{1}_{2^k}\\ 2^k+1,\cdots,2^{k+1} & 1,\cdots,2^k\\ \mathbf{A}_k+2^k\times\mathbf{1}_{2^k} & \mathbf{A}_k\end{bmatrix} \tag{1}$$

直接这么推整个矩阵可能超时，所以写个递归函数直接求矩阵中的某个元素就行了。注意到这是一个尾递归，可以优化成一个 `for` 循环。

时间复杂度　$O(n)$。

Problem E　最后一个

难度　1

关键字　博弈，暴力

题解　这题看上去很像 Nim 问题，但直接当 Nim 做的都 WA 了。注意到数据范围很小，可以记忆化搜索所有可能的状态，求出它们是先手胜还是后手胜，最后查表即可。

那么如果这题的数据范围比较大该怎么做呢？我们可以分类讨论，称 k 为包含 2 个以上石子的堆的个数。如果 $k=0$，那么判一下奇偶问题就解决了。否则，如果 $k=1$，那么 wym6912 可以判断一下奇偶，并决定将这堆大于 1 的石子拿光，或者拿得剩下一个，其中一定有一种必胜策略。

剩下就是 $k>1$ 的情况，这时就能用 Nim 问题的解法了。如果 wym 强行把这题当作 Nim 问题来做，假设他在 Nim 问题中是必胜的，他总可以保证 wang9897 处于必败态，即 SG 函数等于 0。注意到每次操作至多使得 k 减小 1，而终态（只有一堆石子，且这一堆里面有 1 个石子）是 $k=0$ 的，因此在游戏过程中必须经过 $k=1$ 的状态。显然，$k=1$ 的状态的 SG 函数不等于 0，这说明 wym6912 能够保证 wang9897 不经过这些状态，因此只能是 wym6912 经过这些状态。这样，wym6912 就能改用 $k=1$ 时的必胜策略击败 wang9897。

相反，如果 wym6912 在 Nim 游戏的规则下必败，假设他一开始拿光一堆石子，使得 k 变为 1，则 wang9897 可以使用 $k=1$ 的策略击败他。否则，wang9897 就会处于 $k=2$，且 SG 函数不为 0 的状态，能够使用 Nim 游戏的策略击败 wym6912。因此 $k>1$ 时，游戏的结果和 Nim 博弈是一样的。

时间复杂度　暴力法的时间复杂度为 $O(\max\{a_i\}^n)$，使用判断 k 和 SG 函数的方法的时间复杂度为 $O(n)$。

Problem F　背包弹夹平底锅

难度　3.5

关键字　生成函数，容斥原理，快速 Fourier 变换

题解 我们用符号 z 表示一种新的子弹，符号 x 表示一粒子弹，则一种子弹的所有选择方案可以表示为生成函数

$$G_1(z,x) = 1 + z\sum_{k=1}^{\infty}\frac{x^k}{k!} \tag{2}$$

这里除以 $k!$ 是因为不同的 k 个子弹的全排列是 $k!$，1 表示可以不选这种子弹，而相同子弹的排列只有 1 种。学过高数的同学可以立即看出，由于

$$e^x = \sum_{k=0}^{\infty}\frac{x^k}{k!} \tag{3}$$

有

$$G_1(z,x) = 1 + z(e^x - 1) \tag{4}$$

我们把 n 个 G_1 乘起来，就得到使用 n 种（有些可以不选）子弹的生成函数

$$G(z,x) = [G_1(z,x)]^n = [1 + z(e^x - 1)]^n \tag{5}$$

将其展开，其中包含 z^k 和 x^m 的项就对应于恰好有 k 种子弹，m 粒子弹的方案

$$G(z,x) = \sum_{k=0}^{n}\sum_{j=0}^{k}\binom{n}{n-k,j,k-j}z^k(-1)^j e^{x(k-j)} \tag{6}$$

再把指数函数重新展开

$$e^{x(k-j)} = \sum_{i=0}^{\infty}\frac{x^i(k-j)^i}{i!} = 1 + \cdots + \frac{x^m(k-j)^m}{m!} + \cdots \tag{7}$$

我们只关心包含 $z^k x^m$ 的项，有

$$m![z^k x^m]G(z,x) = \sum_{j=0}^{k}\binom{n}{n-k,j,k-j}(-1)^j(k-j)^m \tag{8}$$

这就是对于特定子弹种类数 k 的答案。组合数学学得好的同学可以用容斥原理一眼看出这个式子，但我太菜了不会容斥。然而，直接用这个式子算的话，每算一个 k 需要 $O(k)$ 的时间，总的时间复杂度是 $O(n^2)$，会超时。注意到式 8 可以表示成

$$m![z^k x^m]G(z,x) = n^{\underline{k}}\sum_{j=0}^{k}\frac{(-1)^j}{j!}\times\frac{(k-j)^m}{(k-j)!} \tag{9}$$

即提取和 j 无关的项后和式变成了一个卷积，就能用快速 Fourier 变换在 $O(n\log n)$ 的时间完成卷积，然后再乘上 $n^{\underline{k}}$ 输出即可。

时间复杂度 $O(n\log n)$。

Problem G　小鸟的修路计划

难度　3 [1]

关键字　递推

题解　这题抽象一下就是问 n 个带标号点能形成的无向简单连通图个数 c_n。那么怎么求 c_n 呢？首先，n 个带标号点能形成的无向简单图（无论是否连通）的个数是 $2^{\binom{n}{2}}$（因为有 $\binom{n}{2}$ 对点，每对都可以连边或不连边）。然后考虑把不连通的图都减掉，不连通的图肯定包含若干连通块，设其中节点 1 所在的连通块恰好有 k 个点，那么选取该连通块的方案数为

$$\binom{n-1}{k-1}c_k \tag{10}$$

即先选取该连通块的剩余 $k-1$ 个点，再乘以 k 个点的连通图个数。其余的 $n-k$ 个点肯定不与已经选出的 k 个点连通，但它们相互之间每对都可以连边或不连边，情况数为 $2^{\binom{n-k}{2}}$。把这些情况全部减掉就得到

$$c_n = 2^{\binom{n}{2}} - \sum_{k=1}^{n-1}\binom{n-1}{k-1}2^{\binom{n-k}{2}}c_k \tag{11}$$

直接用这个式子把所有 c_n 递推出来然后直接输出即可。

时间复杂度　预处理 $O([\max\{m\}]^2)$，每组数据 $O(1)$。

Problem H　超长递增序列

难度　1.5

关键字　模拟

题解　根据 k 和 $a[n]$ 的大小关系分类讨论。如果 $k \geq a[n]$，则

$$k \geq a[n] > \sum_{i=1}^{n-1} a[i] \tag{12}$$

这意味着如果不选取 $a[n]$，无论怎样选取其他数，结果都小于 k。因此，必须选 $a[n]$。反之，如果 $k < a[n]$，显然不能选 $a[n]$。这样，就能直接确定选不选 $a[n]$。然后就把问题转化成了 n 比原来小 1 的问题，继续求解就行了。

时间复杂度　$O(n)$。

[1]由于这题在网上题解泛滥，才过了一堆人。

Problem I　两数和

难度　2.5

关键字　枚举

题解　因为答案字典序最小，显然有 $a_1 \le a_2 \le \cdots \le a_n$，否则就能交换一下元素获得字典序更小的解。显然，$a_1 + a_2$ 和 $a_1 + a_3$ 分别是所有 $\binom{n}{2}$ 个和中的最小值和第二小值。此外，我们可以看出，$a_2 + a_3$ 必然小于等于 $a_i + a_j$ $(2 \le i < j)$。所以，$a_1 + a_2, \cdots, a_1 + a_n, a_2 + a_3$ 必然是所有和中前 n 小的值。这样，枚举前 n 个和，假设其中某个和是 $a_2 + a_3$，就能递推出所有的 $a_1, \cdots, a_n$。检查它们是否是正整数即可。

时间复杂度　需要枚举 $O(n)$ 次，每次需要检查 $O(n^2)$ 个数，检查过程中使用 `multiset`，每次操作的时间复杂度是 $O(\log n)$, 故总的时间复杂度是 $O(n^3 \log n)$。

Problem J　垒箱子

难度　无法估计

关键词　暴力，剪枝

题解　这题一看就是个 NP-Hard 的问题，所以只能写暴力搜索。但是暴力搜索的情况数是 $n! \times 6^n$，$n = 30$ 时有 5.86×10^{55} 种，肯定会超时。注意到如果确定了每个箱子的高，那么它们的长宽取向一定是相同的（否则就可以绕高转 $\pi/2$, 得到的解一定不会更差），这样就只有 $n! \times 3^n$ 种情况了，然而这并没有什么用，仍然会超时。我们加上一个很简单的最优性剪枝：尝试将当前尚未堆放的所有箱子直接放在当前最顶端的箱子上，如果能放上去，就取 3 种放法中可行的，高度最高的方案。将这些方案取得的高度都加起来，就是当前状态继续扩展所能得到的高度之和的上限。如果当前高度加上这个上限值仍然小于已知的最优解，那么当前状态怎么扩展都不会得到更优的解，即可剪掉这个分枝。使用这个剪枝后，再卡一卡常数，就能在时限内通过随机数据。

时间复杂度　最坏 $O(n! \times 3^n)$，我不会算期望时间复杂度。

Problem K　签到

难度　3

关键词　计算几何，二次方程组，Newton 迭代

题解 这道题其实就是寻找 3 个球的交点。第一个球就是单位球，第二个和第三个球则需要根据输入确定。它们的球心很好确定，只要把输入的圆心换算成直角坐标即可。它们的半径则需要将输入的球面上的距离 d_2 换算成三维空间中的距离 d_3。可以看出

$$d_3 = 2\sin\left(\frac{d_2}{2r}\right) \tag{13}$$

要求它们的交点，只要连立三个球面方程求解即可：

$$f_i = (x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 - r_i^2 = 0 \quad (i = 1, 2, 3) \tag{14}$$

因为我太菜了，不会解这个方程组，只能使用数值算法，这里选择数值分析教科书上常见的 Newton 迭代法来求解。Newton 迭代法的求解公式为

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \mathbf{x}_{k-1} - \mathbf{J}(\mathbf{x}_{k-1})^{-1}\mathbf{f}(\mathbf{x}_{k-1}) \tag{15}$$

其中向量值函数 $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$，$\mathbf{J}$ 是 Jacobi 矩阵

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix} = 2\begin{bmatrix} x-x_1 & y-y_1 & z-z_1 \\ x-x_2 & y-y_2 & z-z_2 \\ x-x_3 & y-y_3 & z-z_3 \end{bmatrix} \tag{16}$$

由于种种原因（比如 $\mathbf{J}$ 退化、迭代出现循环等），Newton 迭代法未必能收敛到正确的解。所以可以不断随机初始点，迭代一定次数后判断解是否合法，直到找到两个不同的合法解。

时间复杂度 因为我太菜了，分析不出来。