CCPC - Wannafly & Comet OJ 夏季欢乐赛（2019）题解

A. 完全 k 叉树

若想找到最长的一条路径，不妨找出两个离根最远，且没有公共路径的点。假设树共有 m 层，那么左下角的点到根的距离是 m-1，且这个点一定是一个足够远的点。如果最后一层的点的数量大于倒数第二层的点数，则可以取最后一层的最右边的点作为计算最大距离的点，它到根的距离为 m-1。否则取倒数第二层最右边的点作为计算最大距离的点，它到根的距离为 m-2。(取点方法不唯一)由于没有公共路径，所以直接把这两个点到根的距离相加即可。

标程的这种写法中还会有一种情况是最后一层点数为 0，则左下角和右下角的点到根的距离都为 m-2。(当然你可以采取其他方法免除这种情况)

注意 k=1 的 text，此时应当输出 n-1。(如果不特判，在计算树的层数时可能会超时)

时间复杂度$O(\log_k n)$

出题人代码

B. 距离产生美

签到题，首先我们比较当前位置和上一位置的数字，如果距离过近，就需要改变数字。用贪心的思想发现，把当前位置的数字改成一

个很大的值，确保它和相邻的两个数字的距离都足够大，这样的方案是最优的。（因为可以修改的值域是10^{18}，所以一定能找出这样的数字）然后再往后扫描即可。扫描一遍之后就可以输出答案了。

时间复杂度$O(n)$

出题人代码

C. 烤面包片

首先根据范围可得 4!!一定大于10^9（事实上甚至大于10^{18}），所以 4!!!一定能被 mod 整除（n>4 也同理）。那么我们只需要判断 n≤3 的情况了。直接计算即可。如果写一个阶乘函数，然后调用连续三次来获得答案，如：((n!%mod)!%mod)!%mod 可能会无法通过以下的 text：

2 2

（本应输出 0，但三次调用阶乘函数最终会输出 1，因为第一次调用得到 0，之后再调用阶乘函数发现 0！是 1）

一个较好的处理方式是将 n=0，1，2 特判掉，然后预先计算出 3!!为 720，然后用一次 for 循环解决这道题。

出题人代码

D. 茶颜悦色

由题设 k 为正方形边长。首先我们可以通过确定正方形的底边来确定正方形。比如底边是由$[x, y], [x+k, y]$两点连成的边，那么这样的

底边可以确定一个由 $[x,y],[x+k,y],[x,y+k],[x+k,y+k]$四个点组成的正方形。假设我们用左下角的点$[x,y]$来代表这个正方形(下同)。

现在我们来考虑某个点(a, b)可以被怎样的正方形包含。那么我们会发现，左下角的点的 x 坐标满足（$a-k\le x\le a$）且 y 坐标满足（$b-k\le y\le b$）会包含(a, b)这个点。

我们不妨用扫描线的思想来解决这道题。我们从左到右地扫描 x 坐标，并用线段树维护当前 y 坐标上的点数量。即当目前扫描到区间$[a-k,a]$时，点(a, b)对 y 坐标为[b-k, b]中的正方形有贡献。所以当扫描到 a-k 时，将区间[b-k, b]全部+1。当扫描到 a 时，将区间对应地-1。这样就可以描述(a, b)点对题目答案的贡献了。

具体的做法:每个点(a, b)存两遍,一遍存(a-k, b)且置 flag 为 1，代表扫描到此点时要将区间+1，第二遍存(a, b)且置 flag 为-1，代表扫描到此点时要将区间-1。然后按 x 坐标排序，用线段树扫描 x 坐标即可。每次+1 之后区间查询全局最大值，更新答案。

时间复杂度$\mathrm{O}(n\log_2 n)$

验题人代码

E. 飞行棋

由题设骰子面数为 k，首先我们先来考虑当前到终点的距离 d 小于 k 的情况。只有当扔出的点数$\mathrm{Q}==\mathrm{d}$时，可以到达终点，概率为$\frac{1}{k}$。否则都会到达一个新的点d'，且$d'<k$，则相当于还是会有$\frac{1}{k}$的概率到

达终点。每次掷骰子到达终点的概率均为$\frac{1}{k}$，则此时的期望为 k。

现在我们可以假设我们从终点出发，倒着做即可。假设现在到终点的距离为 n，那么只有可能从距离$[\mathrm{n}-\mathrm{k}, \mathrm{n}-1]$的区间掷骰子到此点，且概率均为$\frac{1}{k}$，再加上投掷骰子的动作会使得期望+1。所以用概率 DP 的思想推得公式为 $E_{\mathrm{n}}=1+\frac{1}{\mathrm{k}}\sum_{\mathrm{i}=n-k}^{n-1} E_{\mathrm{i}}$

但是由于 n 过大，线性的递推是无法解决问题的，所以需要用矩阵快速幂优化计算过程来求解此题。

时间复杂度$O(k^3\log_2 n)$

出题人代码

F :三元组

对于一组满足条件的(i, j):

$$2 * \min\left(a_i+a_j, b_i+b_j\right) \leq \max\left(a_i+a_j, b_i+b_j\right)$$

也必然满足:

$$\min\left(\mathrm{a}_{\mathrm{i}}+\mathrm{a}_{\mathrm{j}}, b_i+b_j\right) \leq \max\left(a_i+a_j, b_i+b_j\right)$$

不妨设 $2\left(a_i+a_j\right) \leq b_i+b_j$, 即$\mathrm{a}_{\mathrm{i}}+\mathrm{a}_{\mathrm{j}} \leq b_i+b_j$

此时, 显然不需要考虑$\min\left(a_i+a_j, b_i+b_j\right)=a_i+a_j$这个条件

移项可得 $\left(2 * a_i-\mathrm{b}_{\mathrm{i}}\right)+\left(2 * a_j-\mathrm{b}_{\mathrm{j}}\right) \leq 0$

直接按照$2 * a_i-b_i$升序进行排序, 记数组为P

此时可以发现性质:

若P_i最多只能和P_j匹配, 则P_i+1只能和$[\mathrm{i}+1, \mathrm{j}]$之间的匹配

此时使用滑窗即可计算答案

又因为我们只考虑了$2(a_i+a_j)\le \mathrm{b_i}+\mathrm{b_j}$的情况

再对每个三元组中的$a_i,\mathrm{b_i}$进行交换，计算一次答案即可。

时间复杂度$\mathrm{O(n+nlog_2}\, n)$

出题人代码

G: 篮球校赛

选出每种能力的前 5 个人(很显然,如果一个人某个位置不在前 5，那么他不可能作为此位置出战)，总共选出 25 人（可能重复）

然后使用 DFS 判断答案即可（当然，这一步可以用状压 DP 优化时间，但没必要）

时间复杂度$\mathrm{O}(n+5^5)$

出题人代码

H: 分配学号

先排序，然后判断当前位置的同学的学号有几种选择的方案，累乘获得结果。

时间复杂度$\mathrm{O(nlog_2}\, n)$

出题人代码

I: Gree 的心房

签到题，假设要走的路线是沿着地图的两条边走，那么最优的情况就是这 k 个物品都不挡路,所以判断 k 与(n-1)*(m-1)的关系即可。

时间复杂度 O(1)

出题人代码