Yandex Cup 2017

A. Bicycle Race

题意：给出一个 n 个点 m 条边的无向图，要求找出两个三角形，恰好只有一个公共点，且权值和最大。

题解：先 $O(m\sqrt{m})$ 枚举出所有的三元环，之后枚举公共点，剩下来的问题就是：有 s 个数对 (a_i, b_i)，权值是 c_i，找出两个数对 (a_i, b_i) 和 (a_j, b_j)，使得这 4 个数互不相同，且 $c_i + c_j$ 最大。

这个问题比较简单，考虑枚举出包括 x 的 c 最大的 4 个环，然后这样会得到差不多 $O(4s)$ 个环，然后这些环里面找十几个 c 最大的环，那么答案必定有一个在里面。于是只要随便枚举下另一个环就好了。

B. Those Russian Hackers

题意：有 $k \cdot n$ 秒，标号从 0 到 $k \cdot n - 1$。在第 $i \cdot n + j$ 秒的开始发生系统检查的概率是 p_j，持续时间是一秒。你需要在某一秒的开始登录系统，持续时间是 j $(1 \le j \le 2n-2)$ 的概率是 q_j。中途如果发生系统检查，那么你的任务失败。问最优决策下，你任务成功的概率是多少。

题解：## B. Those Russian Hackers

题意：k 个阶段，每个阶段有 n 秒。每个阶段发生一次系统检查，在第 i 秒发生的概率是 p_i. 你在某一秒登陆系统，持续时间是 j 的概率是 q_j. 中途如果发生系统检查，那么你的任务失败。问最优情况下，任务成功的最大概率。

题解：设 $f(k)$ 表示还剩 k 个阶段的最大概率。当 $k > 1$ 时，在 t 时刻发生检查的概率是 p_t，之后 $(n-t)$ 个时刻都是安全的，那么在 $(t+1)$ 时刻开始登陆成功的概率是 $P(t) = \sum_{i=1}^{n-t} q_i + \sum_{i=n-t+1}^{2n-1} q_i (\sum_{j=i-(n-t)}^{n} p_j)$. 可以使用 FFT 求出每个时间 t 的概率，那么肯定是存在一个时间 t_0，在等待 t_0 的时间后，直接放弃进入下一个阶段。很明显随着 k 增大，t_0 减小。

对于 $k = 0$，类似地也存在一个 t_0，在 t_0 后会直接登录，跟上面情况的唯一区别是 $P(t)$ 不同了。

C. Circular Shift

题意：给出一个字符串 s，问有多少本质不同的字符串 $t = t_1 t_2 ... t_m$，使得 t 和 $t' = t_2 t_3 ... t_m t_1$ 都是 s 的子串。

题解：考虑每一个合法的字符串 $t = cp$，c 是一个字符，p 是一个可以空的字符串，都要满足 pc 也是 s 的子串，所以不妨枚举子串 p，考虑 p 后面接一个字符，前面接一个字符都要在 s 中出现，所以建出 SAM 后，对于每个节点，看他是否有 c 这个后继节点，如果有就看这个子串在头接 c 是否在 s 中出现过，c 可能的位置对应着一段区间。除此之外还要考虑 p 为空和 p 为单个字符的情况。

D. Frog

题意：有 n 个石子，每个石子上一只青蛙。有一个 1 到 n 的排列 $p_1, p_2, ..., p_n$，表示第 i 个石头上的青蛙跳到了第 p_i 块石头。令 a_i 表示有多少个 j 满足 $\min(j, p_j) \le i$ 并且 $i+1 \le \max(j, p_j)$。给出 a 找出一个 p。

题解：a_i 一定是偶数，否则无解。如果 $a_i = 0$，那么显然可以分成两个区间 $[1, i-1]$ 和 $[i+1, n]$，他们之间各自独立。那么假设 a 中没有 0，显然 $a_1 = a_{n-1} = 2$，否则无解。考虑令 $p_1 = n, p_n = 1$，那么把 $[1, n-1]$ 的那些 a 都减去 2，重复上面的检查。

用个线段树就可以模拟上述过程，复杂度 $O(n \log n)$。但是实际上，可以发现，上述过程其实可以用栈实现，这个 $a_1 = a_{n-1} = 2$ 实际就是对应一对匹配括号。

E. HDRF

题意：给一棵有根树，每个节点 u 有个权值, 定义 r_u 是 u 子树内权值的最小值。每次按照下面步骤删掉一个节点：从根出发，如果在节点 u，往 r 最小的那个儿子走一步，如果到了叶子，删掉这个叶子，然后重新计算 r。输出 n 次依次删除的节点。

题解：可以观察到，如果我进入一个子树，那么直到这整个子树删完为止，都不会到其他子树去。那么考虑搞出这个树的欧拉序列，然后建出一个线段树。假设从 u 开始往子树走，由于 u 一定最后走，于是考虑把 u 从线段树中删掉，然后必然是跳到子树内一个权值最小的一个点开始继续走。不断递归搞搞就好了。

F. Counging Orders

题意：给一棵有根树和 v 和 p，求有多少个拓扑序列满足节点 v 在第 p 个位置。

题解：令 $dp(i)$ 表示以 i 为根的子树的拓扑序列个数，假设 u 有 r 个儿子 $v_1, v_2, \ldots, v_r$，且每个儿子对应的子树大小分别是 $k_1, k_2, \ldots, k_r$，那么

$$dp(u) = \binom{k_1 + k_2 + \cdots + k_r}{k_1, k_2, \ldots, k_r} \prod_{i=1}^{r} dp(v_i)$$

考虑从 v 到根的路径中经过的点有 m 个，依次为 $p_1, p_2, \ldots, p_m$，令 $f(i, j)$ 表示在以 p_i 为根的那个子树的拓扑序列中，v 恰好在第 j 个位置。显然，考虑直接枚举 v 在 p_i 那棵子树拓扑序列的位置，然后在枚举这个 v 插入的位置，就可以很轻松的从 $f(i, j)$ 转移到 $f(i+1, j')$。

G. UFO Rectanglars

题意：给一个 $n \times m$ 的矩阵，用边长至少为 3 的空心矩形去画出这个矩阵，要求矩形不能重叠，任意两个矩形不能有两条边都接触。

题解：观察到题目的条件，如果 (x, y) 是矩形的左上角，那么 $(x-1, y-1)$ 一定不是 X。不妨假设我们知道了一个矩形的左上角是 (x, y)，那么考虑所有的 (r, y) $(r > x)$，显然要满足 $(r, y) = X$。如果 $(r, y+1)$ 不是 X，那么这个 (r, y) 显然是边界上的。否则，会有两种情况：

1. $(r, y+1)$ 属于另一个矩形，这个你只需要检查下 $(r+1, y)$, $(r+1, y+1)$, $(r+2, y)$, $(r+2, y+1)$ 和 $(r+1, y+2)$ 就好了。如果是的话，我们找到另一个矩形了，直接递归处理这个矩形。因为，这两个矩形恰好只会有一条边相交，递归进去不会影响外面的这个矩形。
2. $(r, y+1)$ 是当前矩形的下边界，这时候就可以终止枚举了。

对所有的 (x, c) $(y < c)$ 也做同样的处理就好了，复杂度 $O(nm)$。

H. Test For An Intern

题意：给出两个凸包，选定一个移动向量，使得第二个凸包按照这个向量移动后，他们的面积并恰好是 S。

题解：考虑这样一个问题，如果我们能求出两个凸包叠起来的重合的最大面积 M，如果 $S \leq M$，那么显然是可以找到一个向量满足条件的。

首先可以求出一个向量区间，表示沿着这个区间内的向量走可以使得两个凸包相交。那么显然这个最大重合面积关于这个区间是一个凸函数，于是可以在上面三分。之后就是，知道一个凸包沿着某个方向移动，求出最大重合面积。这个是一个经典问题，在 15 年 final 上也出过这个题，也是可以三分搞定的。具体细节参考这个链接

I. Coprime Queries

题意：给一个序列，很多询问 (l, r, x)，问 $[l, r]$ 内下标最大的和 x 互质的数是哪个。

题解：固定 x 可以利用容斥很容易算出和它互质的数量，所以预处理每个数的 square-free 的因子，可以快速回答一段区间内有多少个数是某个 square-free 的倍数。对于每个询问二分答案，算出区间内多少个数和询问的数互质即可。

J. 2084

题意：有 n 队伍，要打若干轮比赛，直到剩下来一个人。如果一个队伍输了 k 次，那么就出局。对于每轮比赛，先按照每个人输的比赛场数分组，相同的分在同一组。对于每一组，假设有 x 人，那么会等概率选出 $\lfloor \frac{x}{2} \rfloor$ 对人进行比赛。如果每个组都只有一个人，那么输的轮数最少的两个队会比赛。现在有个 AI 可以控制 $n-1$ 支队，最后一支队由玩家控制，这支队胜的概率是 p，其他 $n-1$ 支队的结果任意。问玩家胜利的最大概率和最小概率。

题解：注意到状态之和输 x 轮的人数，以及玩家输的轮数有关。这个数量大概比 50 的拆分数小一点，于是可以考虑直接用的 map 存状态来 dp。