Japanese Contest

A. Spanning Trees

题意：给出 n 和 k，要求在 n 个点的完全图中找出 k 个不相交的生成树。

题解：显然可以猜到 k 最大是 $\lfloor \frac{n}{2} \rfloor$，那么考虑下面的构造：假设已经构造出一个偶数 n 的 $\frac{n}{2}$ 个生成树，那么

新增一个节点 $n+1$，那么只要对于第 i 个生成树，添加一条边 $(n+1, i)$，这样就得到了 $\lfloor \frac{n+1}{2} \rfloor$ 个生成树。新增两个节点 $n+1$ 和 $n+2$，首先对于第 i 个生成树，添加两条边 $(n+1, i)$ 和 $(n+2, n-i+1)$，这样就得到了 $\frac{n}{2}$ 个生成树。接下来考虑这样建一个新的生成树：$(n+1, n+2)$，把 $n+1$ 和在区间 $[\frac{n}{2}+1, n]$ 之间的点连边，把 $n+2$ 和在区间 $[1, \frac{n}{2}]$ 之间的点连边。

于是就可以从 $n = 2$ 出发构造出所有的 n 了。

B. Triangle

题意：给出 n 个木棍，长度是 a_i，要求选出 6 个来，构造出两个三角形，使得周长和最长。

题解：显然在数列是 $\mathrm{Fibonacci}$ 数列的时候是最坏情况，已知 $f(75) \geq 10^{15}$，于是显然只要取最大的 200 个差不多就包括了最优解。考虑只选取一个的话，显然就是直接枚举相邻的两个成为短边就可以了。那么对于两个，可以先枚举出一个三角形，另外一个用上面的性质可以 $O(1)$ 算出来。

这边有另一个方法，先从大到小排序，考虑两个三角形分别取了 (a_i, a_j, a_k) 和 (a_u, a_v, a_w) $(i < j < k, u < v < w, k < w)$。

区间 $[i, k]$ 和 $[u, w]$ 不相交，那么显然就是枚举个分界点，然后两边分别取个合法的最大的连续三个数即可。区间 $[i, k]$ 和 $[u, w]$ 相交，那么这 (i, j, k, u, v, w) 6 个数一定是连续的 6 个数。证明的话，随便分析下应该就可以出来了。那么只要枚举连续的六个数，然后暴力 $\binom{6}{3}$ 枚举这些数的划分即可。

C. Boxes and Balls

题意：有 m 个箱子和 n 个球，第 i 个球的重量是 w_i, 初始时所有的箱子都是空的．另有一个长度为 k 的指令序列 a_i, 你要执行 k 次操作，第 i 次操作如下：如果存在一个箱子里包含球 a_i, 那么这轮不用操作, 否则你需要选择一个箱子，把球 a_i 放进去, 并支付代价 w_{a_i}．如果选择了一个已经装有箱子的球，那么你得先把这个箱子清空。求最小的操作代价和。

题解：最小费用流．建立 $k+2$ 个时间点 $0, 1, 2, ...k, k+1$，第 i 个时间点向第 $i+1$ 个时间点连容量 m 费用 0 的边，把指令拆点, 用负费用引流：第 i 个时间点向第 i 个指令的入点连容量 1 费用 w_{a_i} 的边，第 i 个指令的入点向第 i 个指令的出点连容量 1 费用 -10^9 的边，第 i 个指令的出点向时间 $i+1$ 连容量 1 费用 0 的边．跑时间 0 到时间 $k+1$ 的最小费用最大流, 答案是跑出来的费用加上引流用的负边权和．

D. Bit Operations

题意：构造一个由 +, -, * , &, | , ^, ~ 构成的函数 $f(x)$ ，以及 N (≤ 8) 个 pair (x, y)，保证对于每个 pair 都有 $f(x) = y$。

题解：先考虑无解的条件，如果只有位运算，很明显每一位是独立的，每一位只能决定它本身，比如某两个 x_i 的第 k 位是相同的，但他们对应的 y_i 的第 k 位不同则无解。再考虑有了 $+$, - , $*$ 的情况，因为有了 $*$ 就有了左移，也就是低位影响高位的条件。每一位就能决定所有不比它低的位了。也就是说某两个 x_i 的**前** k 位如果是相同的，但他们对应的 y_i 的第 k 位不同则无解。

输入保证有解，所以只需要构造出一个函数 $f_A(x)$，当输入的数 x 后 k 位等于 A 时返回 2^{k-1}，否则返回 0。然后再把这些全部 or 起来即可。

E. Randomized Binary Search Tree

题意：求 n 个点的 treap 深度为 $h(h = 0, 1, 2..n)$ 的概率. 本题中的 treap 是一个随机二叉树, 每个节点有权值和优先级，权值和优先级都是 0 到 1 之间的随机实数. 节点的权值大于它左子树中的所有权值，小于它右子树中的所有权值. 每个节点的优先级都比它所有儿子的优先级要大. treap 的构造方式如下: 先随机选择 n 个权值，然后从空树开始，每次插入一个节点. 插入操作如下: 先随机一个优先级 p, 无视优先级，按照二叉搜索树的方式插入这个节点, 然后考虑优先级，一直把这个节点往上旋转，直到满足了优先级的条件。

题解：由 Treap 的特性可知插入的顺序不影响形态，将节点按照权值排序后最后的树就是由优先级构成的笛卡尔树。设 $f[i][j]$ 为 j 个点构成深度不大于 i 的树的概率，所以有 $f[i][j] = \frac{1}{j}\sum_{k=1}^{j} f[i-1][k-1] * f[i-1][j-k]$。这个是 $O(n^3)$ 的，转移发现是一个卷积可以 FFT，然后再可以发现深度期望是 $O(\log n)$，绝大部分概率集中于期望附近，故只需要算前常数层的概率，复杂度 $O(An\log n)$。

F. Election

题意：有 n 个政党参加选举，总共有 m 个 slot。假设每个政党的得票数是 $c_1, c_2, ..., c_n$，令 $s = c_1 + c_2 + ... + c_n$，那么 * 首先，对每个 i，政党 i 会获得 $\lfloor \frac{c_i}{s} \cdot m \rfloor$ 个 slot; * 然后，剩下来的 slot 会分给 $\frac{c_i}{s} \cdot m$ 小数部分大的那些政党，每个政党一个 slot。如果有多个可以分，编号小的会获得。现在告诉你 n 个政党获得了恰好 $a_1, a_2, ..., a_n$ 的投票，至少获得了 $b_1, b_2, ..., b_n$ 的 slot，问 m 的最小值。

题解：考虑 $m = ks + r$，其中 $r < s$，每个人一开始的 slot 就是 $k \cdot a_i + \lfloor \frac{c_i}{s} \cdot r \rfloor$，那么显然多余的 slot 一定在 r 中被分出去。于是是考虑枚举 r，我们可以计算出当前每个人能够额外获得的 slot，设为 $extra_i$。那么我们就要找到一个 k，使得 $k \cdot a_i + extra_i \geq b_i$ 就好了，可以知道 $k \geq \max\{\lceil \frac{b_i - extra_i}{a_i} \rceil\}$。

G. Hash

题意：给出一个 hash 函数，构造 100 个最长只有 50 个字符的字符串，使得 hash 值一样。

题解：生日攻击，随机生成一群长度为 7 的字符串，从中挑出两个 hash 相同但串不同的字符串，找 7 组，2^7 拼起来即可。

H. K-th String

题意：问有多少长度为 n 的字符串 t，满足：这 n 个字符是前 n 个字符的排列字符串 s 是 t 的一个子串 t 恰好有 $k-1$ 个子串字典序小于 s

题解：由于 t 的每个字符都互不相同，那么显然只要比较首字母就能知道字典序的大小，于是如果知道那些比 $s[1]$ 小的字符都在哪些位置就可以解决这个题了。

考虑枚举字符串 s 的位置，那么可以知道 s 内部比 $s[1]$ 小的字符对答案的贡献。接下来考虑 $dp(i,j,k)$，表示前 i 个未填位置，已经用了 j 个比 $s[1]$ 小的字符，当前总共对答案的贡献是 k 的方案数。转移方程很显然，分这个位置填不填比 $s[1]$ 小的字符即可。

I. Shortest Path Queries

题意：给出 $W\times H$ $(1\le W\le 10, 2\le H\le 10^4)$ 的格子，每个格子上有个非负权值 $A_{x,y}$。给出 Q 个询问，每次给出两个格点，问他们之间的最短路（经过的格子的权值和）。

题解：建出线段树，每个节点对应一个区间，求出中间那一行对于其他内部所有节点的最短路。然后对于每个询问只需要看询问的节点是否在中间那一行的两侧，如果在两侧那么任意两点间的路径都要经过中间那一行，就可以取到答案，否则继续在某侧递归下去。复杂度为 $O(nm\log^2 n+qm\log n)$。

J. Kitamasa's Counterattack

题意：有 n 个箱子，m 把钥匙和 d 个商店. 第 i 把钥匙可以打开 $a_{i,1},a_{i,2},..a_{i,k}$ 这些箱子中的一个. 钥匙被使用过一次后就会消失. 第 i 把钥匙只在商店 s_i 有售，价格为 c_i. 每把钥匙只能被买一次. Akiba 想要买一些钥匙来打开所有的箱子。Kitamasa 为了阻止 Akiba, 他可以在 Akiba 开始购买前，花费 b_i 的代价来把商店 i 的所有钥匙的售价增加 1. Kitamasa 的操作次数没有限制. Akiba 的目标是最小化 $Akiba-Kitamasa$, Kitamasa 的目标是使这个花费差最大化。你需要求出在双方都最优化决策下的 $Akiba-Kitamasa$. 如果答案是无穷大输出 -1. 保证如果 Kitamasa 什么都不做的话 Akiba 可以打开所有的箱子.

题解：令 f_i 表示第 i 把钥匙的购买情况 $(0$ 或 $1)$，λ_i 表示第 i 个商店的加价情况，那么其实就是求：$\max_\lambda \min_{f:matching}\sum_i(c_i+\lambda_i)f_i-\sum_i b_i\lambda_i$ 整理得：$\max_\lambda \min_{f:matching}\sum_i c_if_i+\sum_i(\sum_{j:s_j=i}f_j-b_i)\lambda_i$ 把 λ 视作拉格朗日乘数，则上式的原式为：$\min_{f:matching,\sum_{j:s_j=i}f_j\le b_i}\sum_i c_if_i$ 这怎么看都是最小费用流。

K. Wrapping

题意：给出一个边长为 1 的立方体，你要在上面绑一个环形的绳子，要求：绳子是拉直的给出一个向量 (a,b)，在底面的绳子所在直线必须和它平行题目中有个图，图上两个角必须相同

题解：显然，如果 a 和 b 不互质，可以把最大公约数除掉，于是默认 a 和 b 互质。

考虑你把正方形在平面上无穷无尽地展开，那么绳子肯定是一条直线，并且方向是 (a, b)。然后，想象下这个正方形在平面上沿着这条直线直线滚动（水平或者竖直），那么如果某一次这个正方形的朝向和初始位置一样，那么显然这个就是绳子的长度。

如果把每个面从 1 到 6 标号，那么滚动一次，实际上就是对面的标号做了一次置换，也就是说滚了若干次之后，这个置换变成了原来的。也就是说，如果我们能够求出在一个 $a \times b$ 滚完之后的置换 p，那么这个题就做完了。因为接下来只要找到一个最小的 r，使得 p^r 是 $\{1, 2, 3, 4, 5, 6\}$，那么答案就是 $r \cdot \sqrt{a^2 + b^2}$。

考虑在一个 $a \times b$ 的矩形的滚动状态，可以发现，如果这条直线和 $x = i$（i 是整数）有交，那么就会水平滚一次，否则会垂直滚一次（这个时候肯定和 $y = i$ 有交）。然后，可以发现，这个滚动有一部分是重复出现的。以 8×3 为例，滚动序列是 $xxxyxxxyxxy = x(xxy)x(xxy)(xxy)$，不妨令 $y' = (xxy)$，可以得到 $xy'xy'y'$。也就是说，我们把规模从 8×3 变到了 3×2。

很显然，这个实际上就是一个扩展欧几里得的过程，在扩展欧几里得的同时求出这个置换 p 就可以了。具体过程可以这么考虑：

1. 如果 a 等于 0，那么显然 $p = y$；如果 $b = 0$，显然 $p = x$，这是因为 $\gcd(a, b) = 1$，另一个数肯定是 1。
2. 如果 $a \ge b$，那么可以令 $y' = x^{\lfloor \frac{a}{b} \rfloor} \cdot y$，$(a, b)$ 变成 $(a \bmod b, b)$，继续做。
3. 如果 $a < b$，那么可以令 $x' = x \cdot y^{\lfloor \frac{b}{a} \rfloor}$，$(a, b)$ 变为 $(a, b \bmod a)$，继续做。

L. Ascending Tree

题意：给出一棵有根树，每个节点上有个权值。每次你可以花费 1 的代价加减一某个点的权值。问最小代价，使得父亲的权值严格大于儿子的权值。

题解：首先考虑链上的版本是一个非常经典的问题，出自 BOI2004 的 Sequence，做法是从左到右维护递增序列段，如果遇到一个新元素不满足递归性质就让它和序列的尾合并，做法可以参考此国家集训队论文。

对于树上版本方法类似，只是维护的是和根联通的一个满足性质的连通块，每次取出权值最大的儿子和它的父亲合并，复杂度为 $O(n \log n)$。