A. Determinant

随机 n 个数把矩阵补全成 $n \times n$ 的。
那么就是要算伴随矩阵的第一行，也就是逆矩阵的第一列，高斯消元即可。

B. Roads

考虑 Kruskal 的过程，肯定是同样权值的边连通了一个点集。如果要让 MST 变大，就是要让某个权值的边不再连通。
这是全局最小割问题，可以网络流也可以用 Stoer–Wagner 算法。

C. Intersection

首先高斯消元把 A 和 B 变成线性无关组。
之后就是求方程 $\sum a_i x_i = \sum b_j y_j$ 的解数，再次高斯消元得到零空间的维数 d，答案就是 2^d.

D. Super Resolution

直接输出。

E. Partial Sum

$|\sum_{j=l+1}^{r}|a_j = \max\{S_r - S_l, S_l - S_r\}$，其中 $S_i = a_1 + a_2 + \cdots + a_i$.
所以等价于选择 $2m$ 个前缀和，其中 m 是正号，m 个是负号。
自然是最大的 m 作为正号，最小的 m 个作为负号。
只需要对前缀和排序，枚举 m 并更新答案。

F. Longest Common Subsequence

首先对 a 离散化，之后可以 $O(n^3)$ 枚举序列 X.
如果之后用 $O(n)$ 的 LCS dp 会使复杂度变成 $O(n^4)$.
std 用的方法是 2^3 枚举 X 的一个子序列，通过预处理一个 $next(i, c)$ 表示 i 位置后 c 字符第一次出现的位置，来 $O(1)$ 判断是否是 A 的子序列。

G. Parentheses

括号序列就是要求前 $(2k+1)$ 个里面至少要有 k 个左括号。
那么先把所有括号翻成右括号，之后重新翻回左括号。
那么从左到右贪心，用一个堆维护现在可以翻回左括号的位置。
每次相当于加两个当前段的字符，取一个最小的。所以事件只有最小的被拿完了，或者当前段拿完了。模拟即可。

H. Highway

按照 Prim 算法计算生成树。
假设初始点 v_0 是某条直径的端点。那么距离 v_0 最远的 v_1 必然是直径的另一个端点。
又因为距离任意点最远的要么是 v_0 要么是 v_1，所以剩下的点只需要连往 v_0 和 v_1 中较远的一个即可。

I. Strange Optimization

首先，$\{\frac{i}{n} - \frac{j}{m} : i, j \in \mathbb{Z}\} = \{\frac{\gcd(n,m)k}{nm} : k \in \mathbb{Z}\}$

取 $\frac{1}{2} + \alpha = \frac{\gcd(n,m)}{2nm}$，即可取得最大值 $\frac{\gcd(n,m)}{2nm}$.

J. Similar Subsequence

设 $f(i, j, x, y)$ 表示分别匹配到 a_i 和 b_j，数字的上界和下界分别是 x 和 y 的方案数。注意到 x 和 y 总有一个等于 b_j，所以状态数是 nm^2 的。
转移就是枚举 a_{i+1} 匹配的是 b_k，要求 b_k 落在 $[x, y]$ 中。这个可以用树状数组优化。复杂度是 $O(nm^2 \log m)$.