T1

经过观察可以得到答案就是卡特兰数， $ans = \frac{(2n)!}{n!\cdot(n+1)!}$

T2

从小到大填，然后 $dp[i][j]$ 表示第一行填了 i 个，第二行填了 j 个的方案数。时间复杂度 $O(n^2)$ 。

满分做法需要注意到两个性质：

- 如果 s 全为零或者全为一，那么答案就是卡特兰数（证明可以参考[HNOI2009] 有趣的数列）
- 对于一个 01 切换的节点，会有 $a_i < b_i < b_{i+1} < a_{a+1}$ 。可以得出每个连续相同的部分将 1 ~ $2n$ 分割成了若干个连续的块。

于是答案就是每个连续块大小对应的卡特兰数的乘积。

T3

枚举左端点后直接扫一遍， $O(n^2)$ 可以拿到 50 分。

满分做法有两种

做法一

考虑对每个数字 a_i 用单调栈求出其作为最小值的区间 $[l, r]$ ，然后枚举其中一个端点（假设是左端点），这样根据题意就能确定另外一个端点的值为 $a_l \oplus a_i \oplus D$ ，就变成区间询问一个数字出现了多少次。对每个数字开一个 $vector$ 记录该数字出现在哪些位置，之后二分即可得到答案。在枚举端点时，固定枚举较短的那边可以保证每个位置都只会被最多枚举 $\log n$ 次，于是时间复杂度为 $O(n \log^2 n)$ 。

做法二

分治处理答案，考虑如何合并区间，即计算横跨中点的贡献。

采用双指针，分最小值在左边、在右边两种情况考虑，这两种情况的做法基本一致，这里以最小值在左边为例：

设在左半部分 $[l, mid]$ 的端点为 i ，右半部分 $[mid+1, r]$ 的端点为 j ，那么每次枚举 i 时，可以得出对应的 j 满足，当左端点位于 i 、右端点位于 $[mid+1, j]$ 时，区间的最小值为 $\min_{i \le k \le mid} a_k$ 。

此时，我们需要处理的问题就是在区间 $[mid+1, j]$ 中，有多少 j 满足 $a_i \oplus a_j \oplus mn = D$ ，改写一下变成 $a_i \oplus mn = a_j \oplus D$ ，这样在扫的过程中一直记录 $a_j \oplus D$ 的 cnt 即可，时间复杂度 $O(n \log n)$ 。

T4

采用状压DP，表示当前在哪个点，选了哪些树。时间复杂度为 $O(2^m nm)$ ，可以拿到 30 分。

考虑状压DP转移的过程，可以发现每次新加入一棵树时，若玩家选择进入这棵树，根据当前的局面会有两种游戏规则：

- 若进入到起始点则获胜
- 若进入到起始点则失败

根据两种不同的规则，就会有 $2 \times 2 = 4$ 种不同的胜败结果： $(0,0), (0,1), (1,0), (1,1)$ ，可以把树分成四类。

通过分析可以发现，这 4 种状态分别有如下意义

- 如果只存在这种树，则必败（无论何种规则都无法取胜）
- 如果存在这种树，则必胜（无论其他树的集合对应的是必胜还是必败态，我都可以控制先/后进入起始点）
- 对当前胜负情况不做改变（走完这棵树回到起始点后，先后手不变）
- 对当前胜负情况做出反转（走完这棵树后，交换先后手）

那么先手获胜的情况只有两种

- 存在必胜树
- 没有必胜树，反转树的个数为奇数

根据对应树的数量计算即可